NAVAL
POSTGRADUATE
SCHOOL

MONTEREY, CALIFORNIA

DISSERTATION

AFFINE EQUIVALENCE AND CONSTRUCTIONS OF CRYPTOGRAPHICALLY STRONG BOOLEAN FUNCTIONS

by

Jong H. Chung

September 2013

Dissertation Supervisor: Pantelimon Stanica

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 2013	Dissertation – Jan 09 - Sep 13

4. TITLE AND SUBTITLE: AFFINE EQUIVALENCE AND CONSTRUCTIONS OF CRYPTOGRAPHICALLY STRONG BOOLEAN FUNCTIONS

5. FUNDING NUMBERS

6. AUTHOR(S): Jong H. Chung

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 94942-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES: The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. I.R.B. Protocol number N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

In this thesis, we study a type of affine equivalence for the monomial rotation-symmetric (MRS) Boolean functions and two new construction techniques for cryptographic Boolean functions based on the affine equivalence of cryptographically strong base functions and fast Boolean operations. Affine equivalence of cryptographic Boolean functions presents a formidable challenge to researchers, due to its complexity and size of the search space. We focus on an affine equivalence based on permutation of variables for MRS Boolean functions and their relationship to circulant matrices over the binary field $\mathbb{F}_2$ and regular graphs. We first establish a relationship between generalized inverses of circulant matrices in $\mathbb{F}_2$ and their generating polynomials. We then apply the relationship to gain insight into necessary conditions for the affine equivalence, based on permutations of variables for MRS Boolean functions. We also propose a theoretical connection between regular graphs and MRS Boolean functions to further our study in affine equivalence. Finally, we present two constructions for Boolean functions with good cryptographic properties. The constructions take advantage of two affine-equivalent base functions with strong cryptographic properties. We analyze the cryptographic properties of the constructions and demonstrate an application with these base functions, called the hidden weighted-bit functions.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Boolean Function, Cryptography, Affine Equivalence	175
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

AFFINE EQUIVALENCE AND CONSTRUCTIONS OF CRYPTOGRAPHICALLY STRONG BOOLEAN FUNCTIONS

Jong H. Chung
Major, United States Army
B.S. Applied Mathematics, University of California Los Angeles, Los Angeles, CA, 1996
M.S. Applied Mathematics, Georgia Institute of Technology, GA 2005

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
September 2013

Author:

Jong Ho Chung

Approved By:

Pantelimon Stanica Professor Department of Appl. Math. Dissertation Supervisor	Jon T. Butler Distinguished Professor Department of Elec. and Comp. Eng.
David R. Canright Associate Professor Department of Appl. Math.	Harold Fredricksen Professor Emeritus Department of Appl. Math.
Ralucca Gera Associate Professor Department of Appl. Math.	Craig W. Rasmussen Professor Department of Appl. Math.

Approved By:

Carlos F. Borges, Professor and Chair, Department of Appl. Math.

Approved By:

O. Douglas Moses, Vice Provost for Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In this thesis, we study a type of affine equivalence for the monomial rotation-symmetric (MRS) Boolean functions and two new construction techniques for cryptographic Boolean functions based on the affine equivalence of cryptographically strong base functions and fast Boolean operations. Affine equivalence of cryptographic Boolean functions presents a formidable challenge to researchers, due to its complexity and size of the search space. We focus on an affine equivalence based on permutation of variables for MRS Boolean functions and their relationship to circulant matrices over the binary field $\mathbb{F}_2$ and regular graphs. We first establish a relationship between generalized inverses of circulant matrices in $\mathbb{F}_2$ and their generating polynomials. We then apply the relationship to gain insight into necessary conditions for the affine equivalence, based on permutations of variables for MRS Boolean functions. We also propose a theoretical connection between regular graphs and MRS Boolean functions to further our study in affine equivalence. Finally, we present two constructions for Boolean functions with good cryptographic properties. The constructions take advantage of two affine-equivalent base functions with strong cryptographic properties. We analyze the cryptographic properties of the constructions and demonstrate an application with these base functions, called the hidden weighted-bit functions.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AES	Advanced Encryption Standard
ANF	Algebraic Normal Form
BDD	Binary Decision Diagram
CDMA	Code Division Multiple Access
DES	Data Encryption Standard
GAC	Global Avalanche Criterion
GSM	Global Systems for Mobile Communications
HWBF	Hidden Weighted-Bit Function
LFSR	Linear Feedback Shift Register
MRS	Monomial Rotation Symmetric
OSANF	Ordered Short Algebraic Normal Form
PRBG	Pseudo-Random Bit Generator
RSA	Ron Rivest, Adi Shamir, Leonard Adleman
SAC	Strict Avalanche Criterion
SANF	Short Algebraic Normal Form
S-box	Substitution Box
SPN	Substitution Permutation Network
XOR	Exclusive OR (logical operation)

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGEMENTS

I thank God for leading me through my life. I feel very blessed.

I thank the National Security Agency for the scholarship, which made this journey possible. I also thank the U.S. Army for allowing me to take part in this opportunity.

I express my gratitude to the PhD committee members. I want to thank my advisor Professor Stanica for teaching me everything I know about Boolean functions, inspiring me in the research, being patient, and being a good friend. I would not have been able to do this without all the visits to his office. I want to thank Professor Fredricksen for showing me the light in discrete mathematics, being a wise mentor, and occasionally challenging me with hard but engrossing problems that developed my skills. Working on the problems kept me going in tough times. I want to thank Professor Owen for leading me through the most fundamental math in research, Professor Gera for being an excellent teacher and helping me develop clearer vision in graph theory, Professor Canright for helping me better understand the fundamentals of cryptography. I want to thank Professor Butler for showing me how to navigate the complicated world of parallel computing. I thank Professor Rasmussen for being someone to talk to when I needed and helping me understand the academics at the Naval Postgraduate School. I also want to acknowledge professors Carlos Borges, Frank Giraldo, Wei Kang, and Clyde Scandrett for sound guidance and encouraging words. Numerous other faculty members here at NPS have also helped me overcome various stumbling blocks along the way to this dissertation. Thank you.

I thank my peers and the students at the math department and the post-docs for meaningful conversations, friendship and support. Special thanks to doctors Cesar Aguilar and Eric Choate for sharing their experience and technical expertise.

Finally, I thank my family for accompanying me for the long, adventurous journey. I thank my son, Joseph Minwoo, for all the wonderful things that he is. I thank my wife, Eun Ha, for her prayers, spiritual guidance for the family, and being a wonderful mom and a faithful wife.

THIS PAGE INTENTIONALLY LEFT BLANK

1. INTRODUCTION

As we connect to the Internet with increasing frequency for various services, the need for secure communication is higher than ever before. The ability to email or socialize electronically with the world in a secure and stable manner is crucial for today's global citizen. We want our financial transactions over the Internet to get processed without error. Cyber warfare between nations and industrial espionage among corporations are commonplace. A nation's infrastructure networks need impregnable protection. We are living in a fast moving, networked world, and any compromised or misintended information may result in catastrophic consequences. It is therefore a paramount requirement of every electronic communications network system that it provide every authorized user.

Due to the Internet revolution, the application of cryptography is no longer limited to corporations or government agencies. Any entity on the Internet has the need to protect information in storage and data in transit to another part of the network. This protection, attained via complex (mostly mathematical) schemes called *cryptosystems*, is an integral part of any reliable network service. At the heart of every cryptosystem is a cipher. A cipher is a set of algorithms used to encrypt and decrypt a message. An encrypted message in any language is called ciphertext, and an unencrypted message is called plaintext. In general, there are two types of cryptosystems; asymmetric and symmetric. The security of a modern electronic cipher often depends on secret keys that are essential for encryption and decryption processes. An asymmetric cipher uses different keys to encrypt and decrypt a message, and the connection between the encryption and decryption keys is based upon a known (and well studied) mathematical problem. RSA (the initials of the surnames of its designers, Ron Rivest, Adi Shamir and Leonard Adleman) is a well known asymmetric cipher. Compared to symmetric ciphers, asymmetric ciphers are generally slow. However, asymmetric ciphers have added more functionality, such as message authentication and digital signature and are more efficient in secret-key management, since they require fewer secret keys. A symmetric cipher uses the same secret key to encrypt and decrypt a message.

It is faster than asymmetric cipher, but requires more secret keys, since each pair of users on the network needs to have a unique key. This makes secret-key management a difficult task. Depending on how a symmetric cipher processes a message before encryption or decryption, a symmetric cipher can be further classified into a block or stream cipher. A block cipher breaks down a message into 64, 128, 192 or 256 binary bit blocks and encrypts the message by blocks. The decryption of a block cipher is usually accomplished by reversing the encryption process. Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are well known examples of block ciphers. On the other hand, a stream cipher encrypts and decrypts a bit at a time. For example, GSM (Global System for Mobile Communications), a wireless communications protocol, uses a stream cipher called A5/1.

The subject of this thesis, cryptographic Boolean functions, applies to both ciphers — asymmetric and symmetric. Boolean functions can be key components to hashing algorithms of asymmetric ciphers. Cryptographic Boolean functions can also be an element for block cipher design and analysis. A good illustration of this is DES. Figure 1.1 shows the DES encryption process. Despite all the seemingly complex procedures and diagrams, the only nonlinear component in DES is the substitution process in the function f, which uses a lookup table called substitution box or S-box to simply shuffle data. Surprisingly, in DES, the S-boxes are the only component that integrates significant complexity to the cipher. The S-box is the keystone of the security of DES. The same is true for AES. It is possible to analyze an S-box with cryptographic Boolean functions and measure the security of a block cipher against known attacks. We can also design another set of S-boxes for DES, which optimizes certain cryptographic properties of Boolean functions [1].

The two important qualities of a cipher are security and speed. They often conflict with each other and affect the decision to choose the optimum cryptographic Boolean functions for a cipher. The two broad topics of this thesis are the affine equivalence and construction of Boolean functions with good cryptographic properties. A cryptographic Boolean function of n variables takes an n dimensional Boolean vector and maps it to 0 or 1. Two Boolean functions are affine equivalent if we can obtain one from the other through

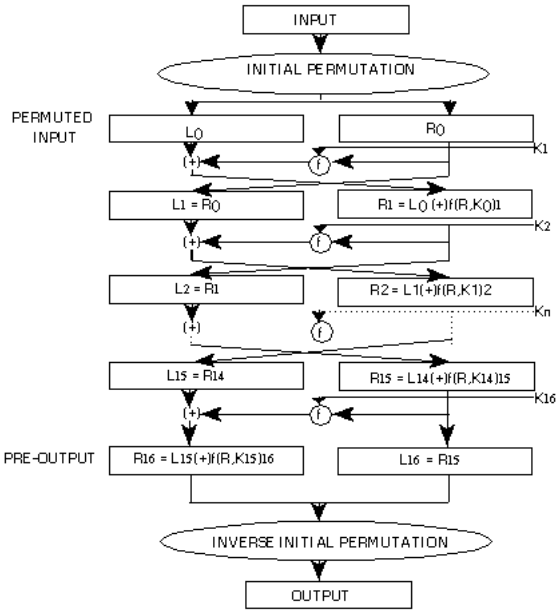


Figure 1.1: Data Encryption Standard (DES) Diagram From [2]

a set of affine transformations. By reflexivity, symmetry, and transitivity, the affine equivalence is an equivalence relation. Therefore, it partitions any set of Boolean functions into equivalence classes. A cryptanalyst can take advantage of the partitioning to devise an efficient algorithm to test the security of a cipher. He needs only to consider the equivalence classes instead of all possible Boolean functions for the cipher, since affine transformations preserve many of the cryptographic properties. On the other hand, cryptographic engineers can integrate affine equivalent functions with good cryptographic properties for speed and simplicity. For example, instead of using the same function, they may use affine equivalence classes of the function to increase security. They can also avoid the equivalence class of a cryptographically weak function, since they are inherently a security risk. Affine equivalence is notoriously complex and often requires unrealistic computing resources. In this thesis, we focus on an affine equivalence of monomial rotation-symmetric (MRS) Boolean functions. A rotation-symmetric Boolean function (RSBF) is a Boolean function such that a Boolean vector and its rotation equivalents render the same function value. For example, if a Boolean function $f(\mathbf{x})$ is a RSBF of three variables $\mathbf{x} = (x_1, x_2, x_3)$, then the vector

$(0,0,1)$ and its rotation equivalents $(1,0,0)$ and $(0,1,0)$ have the same function value. In other words, $f((0,0,1)) = f((0,0,1)) = f((0,0,1))$. RSBFs are well known for their speed [3], and some cryptographically strong Boolean functions are rotation symmetric. An MRS Boolean function is a special type of RSBF, which we formally define in Chapter 4. Construction techniques of cryptographic Boolean functions may be less relevant to the ciphers, such as DES and AES, since they use key-invariant S-boxes. However, ciphers such as BLOWFISH and TWOFISH use key-dependent S-boxes. Efficient construction techniques for S-boxes can be a crucial part of the ciphers with dynamic S-boxes. We study two techniques using affine equivalence of cryptographically strong base functions and two simple Boolean operations, concatenation and complementation. These constructions provide the flexibility to choose a customized base function with good cryptographic properties, as well as speed due to the simplicity of the Boolean operations. We also present an application of our methods, using the hidden weighted-bit function, which is resistant to a binary decision diagram (BDD)-related attack.

The rest of the dissertation is outlined as follows.

In Chapter 2, we formally define basic terminology and principles of cryptographic Boolean functions. We illustrate applications of cryptographic Boolean functions and review common cryptographic properties.

In Chapter 3, we delve into circulant matrices and introduce some results regarding the general inverse of circulant matrices. We study a necessary condition for an affine equivalence based on a permutation of input variables for MRS Boolean functions.

In Chapter 4, we study the relationship between MRS Boolean functions and regular graphs. We establish a basic relationship and suggest other possibilities.

In Chapter 5, we study two different ways to construct Boolean functions with good cryptographic properties via affine transformation, concatenations, and complementations of cryptographically strong base functions.

In Chapter 6, we briefly introduce BDD and cryptanalysis based on its properties. We present an application based on hidden weighted-bit function for our construction methods. We analyze cryptographic properties of these constructions.

In Chapter 7, we summarize and reflect on the main contribution of this thesis. We also suggest some ideas for future research.

THIS PAGE INTENTIONALLY LEFT BLANK

2. CHARACTERISTICS OF CRYPTOGRAPHIC BOOLEAN FUNCTIONS

2.1. BASIC DEFINITIONS AND FUNDAMENTAL PROPERTIES

First, we introduce a commutative binary operation, "exclusive-or" or XOR, denoted by "$\bigoplus$" over the set $\{0,\ 1\}$. The Table 2.1 shows the truth table for the XOR operation.

$\bigoplus$	0	1
0	0	1
1	1	0

Table 2.1: Binary Operation XOR

We also define a multiplication in $\{0,\ 1\}$ in the usual way. This operation is equivalent to logical "AND" operation. The Table 2.2 shows the truth table for the multiplication operation.

$\cdot$	0	1
0	0	0
1	0	1

Table 2.2: Binary Operation $\cdot$

We note that $\{0,\ 1\}$ with $\oplus$ and $\cdot$ forms the smallest Galois field.

Definition 2.1.1. Let the set $\{0,\ 1\}$ with the XOR operation and the usual multiplication be the *binary* or *Boolean field,* denoted by $\mathbb{F}_2$. The set of n-tuples $(x_1, x_2, \ldots, x_n)$, denoted by $\mathbb{F}_2^n$ where $x_i \in \mathbb{F}_2$ with $1 \leq i \leq n$ is an n dimensional vector space over $\mathbb{F}_2$.

We use the terms Boolean vectors and Boolean strings interchangeably. The Boolean vector space has many common properties of other vector spaces, such as $\mathbb{R}^n$ and $\mathbb{C}^n$.

We now proceed to define a Boolean function of n variables.

Definition 2.1.2. We define a Boolean function f of n variables as a mapping

$$f : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2.$$

A Boolean function f takes an n dimensional vector of 1's and 0's as input, and returns 1 or 0 as the function value. We denote the set of all Boolean functions of all variables as $\mathcal{B}$, and the set of all n variable Boolean functions as $\mathcal{B}_n$. We use the terms "Boolean function of n variables" and "Boolean function" interchangeably.

By applying the product rule of combinatorics, we observe that the domain of $f \in \mathcal{B}_n$ has cardinality 2^n. We usually order the domain in a lexicographical order. We distinguish two types of lexicographical ordering, depending on how the elements of the vector domain are ordered. One is the *backward ordering*, where we order the components of the vector $\mathbf{x}$ such that $\mathbf{x} = (x_n, x_{n-1}, \ldots, x_2, x_1)$. Therefore, the domain vectors are lexicographically ordered such that $(0, 0, \ldots, 0, 0)$, $(0, 0, \ldots, 0, 1)$,...,$(1, 1, \ldots, 1, 1)$. The other is the *forward ordering*, where we order the components of the vector $\mathbf{x}$ such that $\mathbf{x} = (x_1, x_2, \ldots, x_{n-1}, x_n)$. Therefore, the domain vectors are lexicographically ordered such that $(0, 0, \ldots, 0, 0)$, $(1, 0, \ldots, 0, 0)$,...,$(1, 1, \ldots, 1, 1)$. When we say "lexicographical order", we mean the backward ordering, unless stated otherwise. For convenience, we regard the vectors as row vectors and use forward ordering unless stated otherwise.

The most popular way to define a Boolean function of n variable is to list the function values as they match the lexicographically ordered domain, which results in a 2^n dimensional Boolean vector or string. The first column of Table 2.3 depicts a Boolean function of 3 variables, $f(\mathbf{x})$ with its truthtable 10011101.

Remark 2.1.3. For convenience, we note that f means the truth table representation of a Boolean function f, and $f(\mathbf{x})$ means the function value at the particular vector $\mathbf{x}$.

Definition 2.1.4. Given a Boolean function f, the complement of f , denoted by $\bar{f}$, is $f \oplus 1$.

We observe that $\bar{f}$ merely flips or changes the function values of f. That is, if $f(\mathbf{x}) = 1$, then $\bar{f}(\mathbf{x}) = 0$, and if $f(\mathbf{x}) = 0$, then $\bar{f}(\mathbf{x}) = 1$. The complement of the function on Table 2.3 is 01100010.

Lemma 2.1.5. *$f \oplus f = \mathbf{0}$, and $f \oplus \bar{f} = \mathbf{1}$ where $\mathbf{0} = (0, 0, \ldots, 0)$ and $\mathbf{1} = (1, 1, \ldots, 1)$.*

Remark 2.1.6. For convenience, we use string and vector notations interchangeably in this thesis. For example, $10011101 = (1, 0, 0, 1, 1, 1, 0, 1)$.

By the product rule of combinatorics, there are 2^{2^n} Boolean functions of n variables. Another operation commonly used in $\mathbb{F}_2^n$ is concatenation.

Definition 2.1.7. Given two Boolean vectors, $f = a_1 a_2 \ldots a_m$ and $g = b_1 b_2 \ldots b_n$ with $a_i,\ b_j \in \mathbb{F}_2$ and m and n in $\mathbb{N}$, the concatenation of f and g, denoted by $f \parallel g$, is an $m + n$ vector obtained by simply combining the elements of f and g in order. That is,

$$f \parallel g = a_1 a_2 \ldots a_m b_1 b_2 \ldots b_n.$$

Example 2.1.8. Table 2.3 shows the various expression of a Boolean function. It is interesting to note that $f = 1001 \parallel 1101$, where $1001,\ 1101 \in \mathcal{B}_2$ and $f \in \mathcal{B}_3$.

Another way to express the truth table is to take -1 to the power of the function value. This set up gives us more options to aggregate some Boolean measures in $\mathbb{R}$.

Definition 2.1.9. Given the truth table of a Boolean function $f(\mathbf{x})$, we define the character form or sign function [4, p. 6] of $f(\mathbf{x})$, denoted by $\hat{f}(\mathbf{x})$

$$\hat{f}(\mathbf{x}) = (-1)^{f(\mathbf{x})}.$$

It is clear that $\hat{f}(\mathbf{x}) \in \{-1,\ 1\}$, and also $\hat{f}(\mathbf{x}) = 1 - 2 \cdot f(\mathbf{x})$.

The second column of Table 2.3 depicts a Boolean function of 3 variables $f(\mathbf{x})$, as $-1, 1, 1, -1, -1, -1, 1, -1$ in sign function. The next lemma describes the relationship between the truth table and the sign function.

Lemma 2.1.10. [4, p. 6] *If $f,\ g \in \mathcal{B}_n$ and $h = f \oplus g$, then $\hat{h} = \hat{f}\hat{g}$.*

We call a multiplication term of Boolean variables, regardless of the power of each variable, a monomial. For example, $x_1 \cdot x_2^0 \cdot x_3 = x_1x_3$ is a monomial. Given $\mathbf{x} = (x_n, \ldots, x_1)$ with $x_i = \{0,\ 1\}$ and $1 \leq i \leq n$, we observe that

$$(x_i)^k = x_i \cdot x_i \cdot \ldots \cdot x_i = x_i,$$

for $k \in \mathbb{N}$. We can write a polynomial-like expression for Boolean functions, using monomials and $\oplus$. When we list the all the possible monomials in lexicographical order, we can regard the set of all the Boolean functions of n variables as the set of the all possible XOR-combinations of n variable monomials. We can also assign a unique 2^n dimensional vector over $\mathbb{F}_2$ to all possible monomials to write an XOR combination of n variable monomials in the following way.

Definition 2.1.11. The algebraic normal form (ANF) of a Boolean function $f(\mathbf{x})$ is an XOR sum of monomials such that

$$f(\mathbf{x}) = \bigoplus_{\substack{\mathbf{a}\in\mathbb{F}_2^n \\ j=1}}^{j=2^n} c_j \cdot x_1^{a_1}x_2^{a_2}\cdots x_n^{a_n},$$

where $\mathbf{a} = (a_1,\ a_2,\ \ldots\ a_n)$, $\mathbf{c} = (c_1, c_2, \ldots, c_{2^n})$, and $a_i,\ c_j \in \mathbb{F}_2$ for $i = 1, 2, \ldots,$ or n and $j = 1, 2, \ldots,$ or 2^n.

Example 2.1.12. The expression below illustrates the ANF of $f(\mathbf{x})$ below. Typically, we order the vector $\mathbf{a}$ lexicographically and obtain binary string $f(\mathbf{x}) = 0001000000001000$ of length 2^n long.

$$
\begin{aligned}
f(\mathbf{x}) &= x_1x_2 \oplus x_3x_4 \\
&= 0 \cdot x_1^0x_2^0x_3^0x_4^0 \oplus 0 \cdot x_1^1x_2^0x_3^0x_4^0 \oplus 0 \cdot x_1^0x_2^1x_3^0x_4^0 \oplus 1 \cdot x_1^1x_2^1x_3^0x_4^0 \oplus 0 \cdot x_1^0x_2^0x_3^1x_4^0 \oplus \cdots \\
&\quad \cdots \oplus 1 \cdot x_1^0x_2^0x_3^1x_4^1 \oplus \ldots \oplus 0 \cdot x_1^1x_2^1x_3^1x_4^0 \oplus 0 \cdot x_1^1x_2^1x_3^1x_4^1.
\end{aligned}
$$

We also note that the ANF of a Boolean function is unique.

A Boolean function may be better understood with one expression type of $f(\mathbf{x})$ than another. We transform an ANF of a Boolean function $f(\mathbf{x})$ to the truth table of $f(\mathbf{x})$ by simply evaluating the function value with the ANF. We can transform a truth table in Table 2.3 into an ANF expression by adding the monomials derived by the input values $\mathbf{x}$ such that $f(\mathbf{x}) = 1$. We demonstrate this process in the next example.

Example 2.1.13. The truth table of the Boolean function, $f(\mathbf{x})$ on Table 2.3 is 10100111, where $f(000) = f(010) = f(101) = f(110) = f(111) = 1$. We construct each term to ensure that $f(\mathbf{x}) = 1$ whenever $\mathbf{x}$ happens to be one of the vectors listed. For example, since $f(011) = 1$, we want to have the term $x_1x_2(x_3 \oplus 1)$ for $x_1 = 1,\ x_2 = 1,\ x_3 = 0$. And we apply this to each $\mathbf{x}$ with $f(\mathbf{x}) = 1$ to obtain

$$
\begin{aligned}
f(\mathbf{x}) &= (x_3 \oplus 1)(x_2 \oplus 1)(x_1 \oplus 1) \oplus (x_3 \oplus 1)x_2x_1 \oplus x_3(x_2 \oplus 1)x_1 \\
&\quad \oplus x_3x_2(x_1 \oplus 1) \oplus x_3x_2x_1 \\
&= 1 \oplus x_1 \oplus x_2 \oplus x_1 \cdot x_3 \oplus x_1 \cdot x_2 \cdot x_3.
\end{aligned}
$$

$n=3$	$f(\mathbf{x})$	$\hat{f}(\mathbf{x})$	$ANF(f(\mathbf{x}))$
000	1	-1	1
001	0	1	1
010	0	1	1
011	1	-1	0
100	1	-1	0
101	1	-1	1
110	0	1	0
111	1	-1	1

Table 2.3: Various Representation of a Boolean function $f(\mathbf{x})$

There is a more efficient way to construct the ANF from the truth table (and vice versa), called *transeunt triangle*, and we refer to [5].

Definition 2.1.14. The ANF of a Boolean function gives us some important measures on the function. In an ANF, the number of variables in the highest-order monomial with nonzero coefficient is called the *degree* of the Boolean function. A Boolean function is *homogeneous* if all its ANF terms have the same degree. A Boolean function is *nonhomogeneous* if it is not *homogeneous*.

Example 2.1.15. The function in Example 2.1.12 is a homogeneous Boolean function with degree 2, whereas the function below is a *nonhomogeneous* Boolean function with degree 5.

$$f(\mathbf{x}) = x_1x_2 \oplus x_1x_2x_3x_4x_5.$$

The degree of a Boolean function is one of the most important cryptographic properties in a cipher. We discuss the cryptographic implications of the degree in the next section. A Boolean function of degree "at most, one" is an *affine* function. An affine function with the constant term equal to zero is called a *linear* function. The set of all n variable affine (respectively linear) functions is denoted by $\mathcal{A}_n$ (respectively $\mathcal{L}_n$).

Let $f \in \mathcal{B}_n$ and E be any flat (that is, a coset of a vector subspace). If the restriction $f|_E$ of f to E is constant (respectively affine), then E is called a *constant* (respectively affine) *flat* for f.

Let

$$1_f = \{\mathbf{x} \in \mathbb{F}_2^n | f(\mathbf{x}) = 1\}$$

be the support of a Boolean function f. We define the complement of the support

$$0_f = \{\mathbf{x} \in \mathbb{F}_2^n | f(\mathbf{x}) = 0\}.$$

We also note the usual dot-product operation of two vectors in the context of Boolean vectors. Let $\mathbf{x} = (x_n, \ldots, x_1)$ and $\mathbf{w} = (w_n, \ldots, w_1)$ both belonging to $\mathbb{F}_2^n$ and $\mathbf{x} \cdot \mathbf{w} = x_n w_n \oplus \ldots \oplus x_1 w_1$.

Definition 2.1.16. The number of 1's in a binary string or vector $\mathbf{x}$ denoted by $wt(\mathbf{x})$, is called the *Hamming weight*.

We can apply the same idea to the truth table of a Boolean function f. The Hamming weight of f is the Hamming weight of the truth table of f. The Hamming weight of the Boolean function on Table 2.3 is 5. We also observe that the cardinality of 1_f is the Hamming weight of f.

Lemma 2.1.17. *Given $f \in \mathcal{B}_n$,*

$$wt(f) = \sum_{\mathbf{x} \in \mathbb{F}_2^n} f(\mathbf{x}) = \left(2^n - \sum_{\mathbf{x} \in \mathbb{F}_2^n} \hat{f}(\mathbf{x}) \right).$$

Definition 2.1.18. Given two binary vectors (or strings) of same length, $\mathbf{x} = (x_1, x_2, \ldots x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots y_n)$. The *Hamming distance*, denoted by $d(\mathbf{x}, \mathbf{y})$, between the two vectors is the number of indices where they have different binary values.

For example, if $\mathbf{x} = (0, 1, 0,0,0,0,0)$ and $\mathbf{y} = (1,1,1,1,1,1,0)$, $d(\mathbf{x}, \mathbf{y}) = 5$ since the elements of $\mathbf{x}$ and $\mathbf{y}$ are different in the indices 1, 3, 4, 5, 6.

Lemma 2.1.19. *Given two Boolean functions of n variables $f = x_1, x_2, \ldots x_k$ and $g = y_1, y_2, \ldots, y_k$ in truth table, $d(f, g) = wt(f \oplus g)$.*

Lemma 2.1.20. *For two Boolean functions f and g,*

$$d(f,\, g) = 2^{n-1} - \frac{1}{2}\hat{f}\cdot\hat{g}.$$

Next, we introduce an important measure of Boolean functions.

Definition 2.1.21. [4, p. 7] Given a Boolean function f, the *Walsh transform* of f on a vector $\mathbf{w}$ is an integer value function defined by

$$W(f)(\mathbf{w}) = \sum_{\mathbf{x}\in\mathbb{F}_2^n} f(\mathbf{x})(-1)^{\mathbf{w}\cdot\mathbf{x}}.$$

We can recover f by the inverse Walsh transform,

$$f(\mathbf{x}) = \frac{1}{2^n}\sum_{\mathbf{x}\in\mathbb{F}_2^n} W(f)(\mathbf{w})(-1)^{\mathbf{w}\cdot\mathbf{x}}.$$

Another way to measure a Boolean function is the Walsh transform of $\hat{f}$ on $\mathbf{w}$, denoted by $W_f(\mathbf{w})$. We refer to it as the Walsh–Hadamard transform of $f(\mathbf{x})$.

$$\begin{aligned} W_f(\mathbf{w}) &= \sum_{\mathbf{x}\in\mathbb{F}_2^n} \hat{f}(\mathbf{x})(-1)^{\mathbf{w}\cdot\mathbf{x}} \\ &= \sum_{\mathbf{x}\in\mathbb{F}_2^n} (-1)^{f(\mathbf{x})+\mathbf{w}\cdot\mathbf{x}} \end{aligned}$$

The Walsh transform of f on $\mathbf{w}$ essentially measures the Hamming distance between f and the linear function defined by the vector $\mathbf{w}$, which is

$$\mathbf{w}\cdot\mathbf{x} = w_1x_1 \oplus w_2x_2 \oplus \cdots \oplus w_nx_n.$$

We use this result to define the nonlinearity of a Boolean function in the next section.

Next, we discuss a concept analogous to a "directional derivative" [4, p. 38]. Given a Boolean function $f(\mathbf{x})$ and an arbitrary vector $\mathbf{u}$, we can consider a measure on $f(\mathbf{x})$ with respect to a vector $\mathbf{u}$.

Definition 2.1.22. Given a Boolean function f, the derivative of f with respect to a vector $\mathbf{u}$, denoted by $D_{\mathbf{u}}f$, is defined by

$$D_{\mathbf{u}}f = f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \mathbf{u}).$$

If $f(\mathbf{x}) = f(\mathbf{x} \oplus \mathbf{u})$, $D_{\mathbf{u}}f = 0$. If $f(\mathbf{x}) \neq f(\mathbf{x} \oplus \mathbf{u})$, $D_{\mathbf{u}}f = 1$. Therefore, $\sum_{\mathbf{x} \in \mathbb{F}_2^n} D_u f(\mathbf{x})$ counts the number of input values in which function values change when the change in direction of u is applied. We can apply the same idea to $\hat{f}$ and obtain $D_{\mathbf{u}}\hat{f} = \hat{f}(\mathbf{x})\hat{f}(\mathbf{x} \oplus \mathbf{u})$, so that $D_{\mathbf{u}}\hat{f} \in \{-1, 1\}$. When we aggregate $D_{\mathbf{u}}\hat{f}$ over $\mathbf{x} \in \mathbb{F}_2^n$, we have the following definition for measuring how sensitive a Boolean function is in the domain.

Definition 2.1.23. [4, p. 8] The *autocorrelation function* of $f \in \mathcal{B}_n$ with respect to $\mathbf{u} \in \mathbb{F}_2^n$, denoted $C_f(\mathbf{u})$ is defined by

$$\begin{aligned} C_{\hat{f}}(\mathbf{u}) &= \sum_{\mathbf{x} \in \mathbb{F}_2^n} \hat{f}(\mathbf{x}) \cdot \hat{f}(\mathbf{x} \oplus \mathbf{u}) \\ &= \sum_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \mathbf{u})}. \end{aligned}$$

We note that $C_{\hat{f}}(\mathbf{0}) = 2^n$.

The autocorrelation function measures the overall change of f as a result of the shift or change caused by a vector u in the domain. We argue that if the overall change is half of 2^n, the statistical impact of the shift of $\mathbf{u}$ is zero. This notion gives us a cryptographic property called the *strict avalanche criterion* (SAC), a concept invented by Webster and Tavares and published in *Crypto 85*, which we elaborate in the next section. We can apply

a similar idea to the autocorrelation function of two Boolean functions and measure how they are related to each other with respect to a vector.

Definition 2.1.24. [4, p. 8] The correlation between two Boolean functions f and g is defined by

$$C(f,\, g) = 1 - \frac{d(f,\, g)}{2^{n-1}}.$$

The correlation function between f and g with respect to $\mathbf{u} \in \mathbb{F}_2^n$ is an integer valued function defined by

$$C(\hat{f},\, \hat{g})(\mathbf{u}) = \sum_{\mathbf{x} \in \mathbb{F}_2^n} \hat{f}(\mathbf{x})\hat{g}(\mathbf{x} \oplus \mathbf{u}).$$

S-boxes of block ciphers may employ multiple cryptographic Boolean functions. We want to reduce the correlation between functions as well as the autocorrelation function values of each function used, to minimize the risk of a correlation attack.

The concept of a derivative gives us another interesting measure of a cryptographic function, namely *linear structure*.

Definition 2.1.25. [6], [7] If the derivative of $f \in \mathcal{B}_n$ in respect to the $\mathbf{u} \in \mathbb{F}_2^n$, $D_{\mathbf{u}}f$ is constant, then $\mathbf{u}$ is a linear structure of f. If the linear structures of f form a subspace in $\mathbb{F}_2^n$, we call this subspace a *linear space* of f.

Depending on the constant derivative, we can further classify a linear structure $\mathbf{u}$ into two types $0{-}linear\ structure$, denoted by $LS_0(f)$ if $D_{\mathbf{u}}f = 0$, and $1{-}linear\ structure$, denoted by $LS_1(f)$ if $D_{\mathbf{u}}f = 1$.

Theorem 2.1.26. [8] *If $LS_1(f) \neq \phi$, the dimension of the entire linear space of f is equal to*

$$\dim(LS_0(f)) + 1.$$

In [9], the concept of linear structure was used to show that the strict avalanche criterion is local in the sense of a derivative, and may not be enough to protect a block cipher from a statistical attack.

2.2. APPLICATION OF CRYPTOGRAPHIC BOOLEAN FUNCTIONS

In this section, we briefly comment on some applications of cryptographic Boolean functions. Boolean functions are typically used for the construction of S-boxes for block ciphers, nonlinear filters for a linear-feedback shift register (LFSR), nonlinear combiners for multiple LFSRs in a stream cipher, or hashing functions in an asymmetric cipher.

2.2.1. Block Ciphers

A block cipher breaks down the text into blocks of some size, and enciphers and deciphers it block by block. Boolean functions play a crucial role in analyzing and designing block ciphers. The two prominent techniques to design a block cipher are Feistel ciphers and substitution permutation networks (SPNs). Regardless of the scheme, it uses substitution boxes or S-boxes. For example, DES uses eight fixed S-boxes, which convert a six-bit input string to a four-bit string. Table 2.4 shows the first S-box of DES, which consists of four lookup tables numbered 0 through 15. Each row can be represented by a vectorial Boolean function, $F(\mathbf{x}) : \mathbb{F}_2^4 \rightarrow \mathbb{F}_2^4$, which can be composed with four four-variable Boolean functions. Each function takes a six-bit string and extracts the first and the last bit to determine which row of the table to use. Then, the middle four bits process through the vectorial function to output the substitution value. Table 2.5 shows the Boolean representation of the first S-box, and Table 2.6 lists the four cryptographic Boolean functions for the first row of the first S-box.

Typically, S-boxes are the only nonlinear features in a block cipher. Without nonlinear S-boxes, almost all block ciphers could be solved with little effort. Therefore, when designing an S-box for a block cipher, we must consider known relevant cryptographic characteristics of S-boxes to optimize their security. In [1], a complete set of replacement S-boxes for DES based on Boolean functions is presented.

Row\Col	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Table 2.4: 1st S-box of DES in Decimal From [4, p. 170]

Row\Col	0000	0001	0010	0011	0100	0101	0110	0111
00	1110	0100	1101	0001	0010	1111	1011	1000
01	0000	1111	0111	0100	1110	0010	1101	0001
10	0100	0001	1110	1000	1101	0110	0010	1011
11	1111	1100	1000	0010	0100	1001	0001	0111

Row\Col	1000	1001	1010	1011	1100	1101	1110	1111
00	0011	1010	0110	1100	0101	1001	0000	0111
01	1010	0110	1100	1011	1001	0101	0011	1000
10	1111	1100	1001	0111	0011	1010	0101	0000
11	0101	1011	0011	1110	1010	0000	0110	1101

Table 2.5: 1st S-box of DES in Binary

Col	Boolean Function (ANF and Truth Table)
1	$1 \oplus x_1 \oplus x_3 \oplus x_4 \oplus x_2x_3 \oplus x_3x_4 \oplus x_1x_2x_3 \oplus x_2x_3x_4$ 1010011101010100
2	$1 \oplus x_3 \oplus x_4 \oplus x_1x_2 \oplus x_1x_3 \oplus x_2x_4 \oplus x_1x_2x_4$ 1110010000111001
3	$1 \oplus x_1 \oplus x_2 \oplus x_1x_2 \oplus x_1x_3 \oplus x_2x_3 \oplus x_1x_4 \oplus x_2x_4 \oplus x_3x_4 \oplus x_1x_3x_4 \oplus x_2x_3x_4$ 1000111011100001
4	$x_2 \oplus x_4 \oplus x_1x_3 \oplus x_1x_4 \oplus x_1x_2x_4$ 0011011010001101

Table 2.6: Boolean Function Representation of the First Row of the First S-box of DES

The S-boxes in DES are predetermined and typically implemented as a lookup table for simplicity. However, block ciphers, such as BLOWFISH [10] and TWOFISH [11], do not use fixed lookup tables (S-boxes), since they generate S-boxes from the key for each session.

2.2.2. Stream Ciphers

A stream cipher encrypts a plaintext bit by bit with secret-key stream bits. In general, an XOR operation of a plaintext bit and secret-key stream bit results in a ciphertext bit. A stream cipher integrates pseudo-random bit generators (PRBG) to produce a key stream. In electronic circuits, a shift resister is a sequential logic circuit for storage of binary data. It is set up in a linear fashion such that the stored data is shifted to a predetermined direction when the circuit is on. A linear-feedback shift register (LFSR) is a shift register which takes the output of a linear function of two or more bits from its previous state [4, p. 19]. We assume an LFSR has $n \geq 1$ variables. Table 2.7 shows the LFSR sequence generated by the Boolean function of 4 variables, $x_1 \oplus x_4$ with the initial vector $\mathbf{x} = x_1x_2x_3x_4 = 0101$. For example, from the initial vector, $x_1 = 0$ and $x_4 = 1$. Therefore, $x_1 \oplus x_4 = 0 \oplus 1 = 1$. This feedback sets the next $x_1 = 1$, and the previous x_1, x_2, and x_3 shift to x_2, x_3, and x_4, respectively, which sets the next state, $\mathbf{x} = x_1x_2x_3x_4 = 1010$. It repeats this process until the LFSR obtains the initial vector again. The number of steps needed to reach the initial vector is called the cycle of an LFSR. We note that the LFSR on Table 2.7 has a cycle of $2^4 - 1 = 15$, which is the maximum cycle possible.

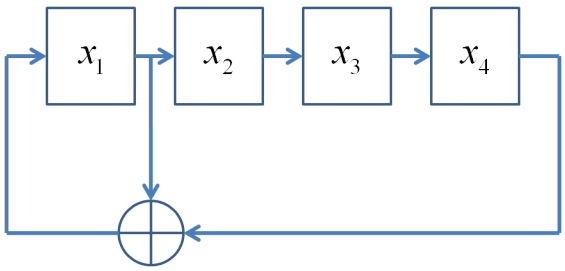


Figure 2.1: LFSR of $x_1 = x_1 \oplus x_4$

We can integrate a nonlinear filter or an n variable Boolean function with good cryptographic properties to generate secure key streams.

One way to construct a PRBG is to combine LFSRs and cryptographic Boolean functions. We consider two applications of cryptographic Boolean functions in stream ciphers: a nonlinear filter and a nonlinear combiner. In the nonlinear filter setup, an LFSR and a cryptographic Boolean function as a nonlinear filter can generate a secret-key stream.

x_1	x_2	x_3	x_4	Output	x_1	x_2	x_3	x_4	Output
0	1	0	1	1	0	0	0	1	1
1	0	1	0	1	1	0	0	0	1
1	1	0	1	0	1	1	0	0	1
0	1	1	0	0	1	1	1	0	1
0	0	1	1	1	1	1	1	1	0
1	0	0	1	0	0	1	1	1	1
0	1	0	0	0	1	0	1	1	0
0	0	1	0	0	0	1	0	1	1

Table 2.7: Bit Stream Generated by LFSR of $x_1 = x_1 \oplus x_4$ with Initial Vector 0101

As the LFSR shifts through the states, the nonlinear filter processes n variables from each state and outputs a key bit. Table 2.2 illustrates this process.

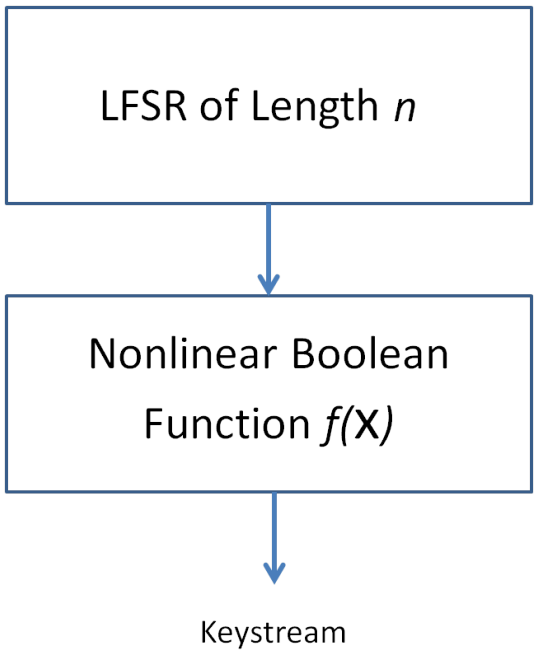


Figure 2.2: Nonlinear Filter

Turing is a stream cipher developed for CDMA (Code Division Multiple Access), which is a wireless communication protocol developed by Qualcomm [12]. Turing generates 160 bits of output in each round by applying a nonlinear filter to the internal state of an LFSR [13]. In the nonlinear combiner setup, an n variable Boolean function with good

cryptographic properties takes n output bits, each from n distinct LFSRs, and outputs a secret stream bit. Figure 2.3 illustrates a nonlinear combiner of n LFSRs. An example for this setup is A5/2, which is the stream cipher used to encrypt voice transmissions in the GSM cellular telephone network. A5/2 is based on four LFSRs and a nonlinear combiner.

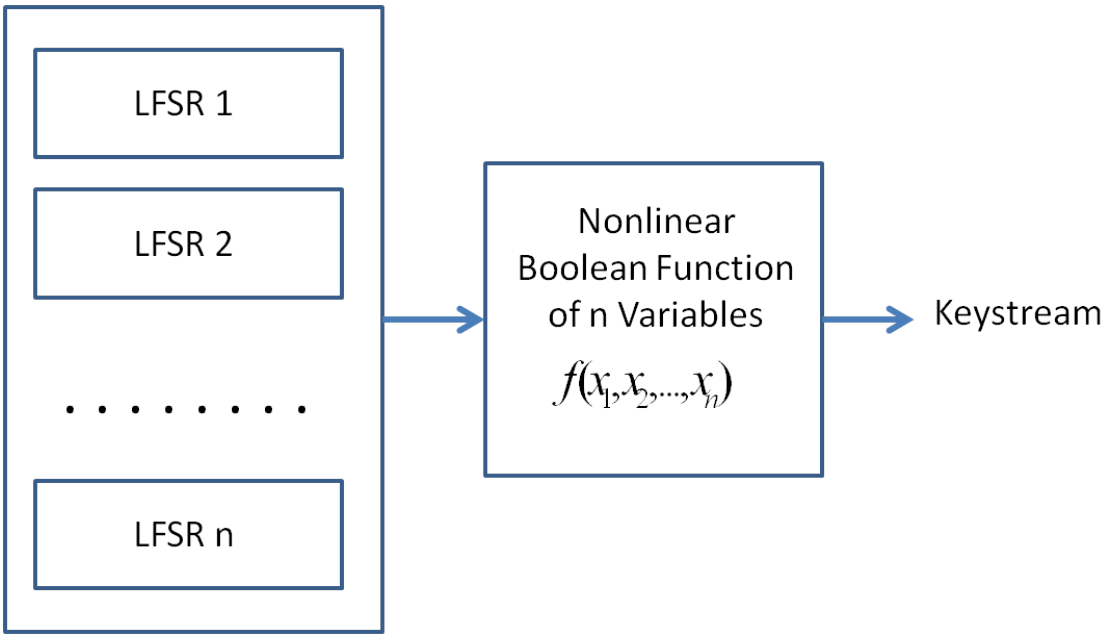


Figure 2.3: Nonlinear Combiner

2.2.3. Hash Functions

Some secure communications protocols and asymmetric ciphers use hash functions to ensure authenticity, integrity, and nonrepudiation of a message. A hashing function can be integrated into a secure communication system to detect an unauthorized modification or tampering. Secure email systems can employ a digital-signature scheme that uses hashing functions to ensure the reliability of a message. Since a hashing function does not require a decryption or recovery of the original message, in a software-based implementation we can use a fast Boolean function with good cryptographic properties. Some candidates for this purpose are symmetric and rotation-symmetric Boolean functions, since we can evaluate them faster due to their simple structures. A Boolean function is symmetric if vectors with

$\mathbf{x}$	0000	0001	0010	0011	0100	0101	0110	0111
$f(\mathbf{x})$: Symmetric	0	1	1	0	1	0	0	1
$g(\mathbf{x})$: RSBF	1	1	1	0	1	1	0	1
$\mathbf{x}$	1000	1001	1010	1011	1100	1101	1110	1111
$f(\mathbf{x})$: Symmetric	1	0	0	1	0	1	1	0
$g(\mathbf{x})$: RSBF	1	0	1	1	0	1	1	1

Table 2.8: Comparison of a Symmetric and Rotation-Symmetric Boolean Function

the same Hamming weight have the same function value. A Boolean function is rotation symmetric if the function renders the same function value for an input vector and its rotation equivalents.

Table 2.8 illustrates the symmetric and rotation-symmetric functions. The function $f(\mathbf{x})$ is symmetric, since has the same function values for the vectors with each Hamming weight. The function $g(\mathbf{x})$ is rotation symmetric, since each vector and its rotation equivalents have the same function values. We note that if a function is symmetric, then it is also rotation symmetric. However, the converse of the previous statement is not true, since a rotation equivalent of a vector with a Hamming weight k and a non-rotation equivalent of the vector with the same Hamming weight may have different function values in a rotation-symmetric function. We give a proper definition of rotation-symmetric Boolean functions and their properties in the next chapter.

2.3. CRYPTOGRAPHIC CHARACTERISTICS OF BOOLEAN FUNCTIONS

In [14], Shannon establishes two important principles in designing a cipher: confusion and diffusion. He introduces the principle of confusion to ensure that the relationship between the ciphertext and the encryption or decryption key is complex and complicated as possible, and the principle of diffusion to ensure the plaintexts are dissipated into the space of ciphertext. Most cryptographic characteristics discussed here are well studied and address Shannon's confusion and diffusion principles in a cipher. We review some well-studied characteristics and outline significance of the corresponding property.

2.3.1. Balancedness

A Boolean function $f \in \mathcal{B}_n$ is *balanced* if the truth table of f has 2^{n-1} zeros and 2^{n-1} ones. We observe that if f is balanced $wt(f) = 2^{n-1}$. A balanced Boolean function counters statistics-based attacks and correlation attacks. We can measure how close the Boolean function is to a balanced one by the following measure.

Definition 2.3.1. [15] The *imbalance* of Boolean function I_f is defined as follows

$$I_f = \sum_{\mathbf{x} \in \mathbb{F}_2^n} \hat{f}(\mathbf{x}).$$

The correlation between $f(\mathbf{x})$ and the constant function $f(\mathbf{x}) = 0$ or 1 is $-1 \leq \frac{I_f}{2^n} \leq 1$. A balanced function f has zero correlation to a constant function, since $I_f = 0$. The balancedness can be checked by the Walsh–Hadamard transform as shown in the lemma below.

Lemma 2.3.2. *A Boolean function f is balanced if and only if $W_f(0) = 0$.*

2.3.2. Algebraic Degree

Consider a Boolean function in ANF, $f(\mathbf{x}) = \bigoplus_{\substack{\mathbf{a} \in \mathbb{F}_2^n \\ j=1}}^{j=2^n} c_j \cdot x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$ as in Definition 2.1.11. The algebraic degree of $f(\mathbf{x})$ is the largest number of variables in a term $c_j \cdot x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}$ with $a_i = c_j = 1$ with $i = 1, 2, \ldots n$. We denote the algebraic degree of $f \in \mathcal{B}_n$ as $\deg(f)$. Using interpolation cryptanalysis [16] and high-order differential cryptanalysis [17], a cryptanalyst can carry out an effective attack on some ciphers employing low-degree Boolean functions.

2.3.3. Nonlinearity

The use of affine Boolean functions in a cipher is undesirable, due to the simple algebraic structure of affine functions. We want to use Boolean functions that are far away from an affine function, which gives us the following measure.

Definition 2.3.3. [4, p. 7] Let $\mathcal{A}_n$ be a set of all affine Boolean functions of n variables. The *nonlinearity* of a Boolean function, denoted by $nl(f)$ is the minimum Hamming distance between f and any function in $\mathcal{A}_n$.

Theorem 2.3.4. [4, p. 13] *For $f \in \mathcal{B}_n$,*

$$nl(f) = 2^{n-1} - \frac{1}{2} \max_{\mathbf{u} \in \mathbb{F}_2^n} |W_f(\mathbf{u})|,$$

The following upper limit for the nonlinearity is well known (see Seberry and Zhang [18]).

Theorem 2.3.5. [18] *For $f \in \mathcal{B}_n$,*

$$nl(f) \leq 2^{n-1} - 2^{n/2-1}.$$

We observe that $2^{n/2-1}$ in Theorem 2.3.5 is not an integer if n is odd. If n is even, we have a special family of functions, called *bent functions*, that achieve the nonlinearity bound.

Definition 2.3.6. Let $f \in \mathcal{B}_n$ and n be even. Then f is a *bent function* if

$$nl(f) \leq 2^{n-1} - 2^{n/2-1}.$$

If n is odd with $n = 2k + 1$, $k = 0,\ 1,\ 2, \ldots$, the *bent concatenation bound* is defined as

$$2^{2k} - 2^k.$$

It is known that the algebraic degree of a bent function is bounded above by $\frac{n}{2}$ [4, p. 80]. The *r-order nonlinearity*, denoted by $nl_r(f)$, is its distance from the set of all n variable functions of algebraic degrees at most r. A Boolean function needs to have higher r-order nonlinearity to resist a fast algebraic attack [19]. We can also devise a statistical measure using nonlinearity.

Definition 2.3.7. Given a Boolean function f, the *bias of nonlinearity* for f, denoted by $\epsilon(f)$ is

$$\epsilon(f) = \frac{1}{2} - \frac{nl(f)}{2^n}.$$

The fast correlation attack on f has an on-line complexity proportional to $\left(\frac{1}{\epsilon}\right)^2$ [20].

2.3.4. Avalanche and Propagation Criteria

2.3.4.1. Strict Avalanche Criterion (SAC)

The strict avalanche criterion is one of the cryptographic characteristics that cover the diffusion principle. The main point is that when we change an element of the input vector, we want the effect of the change equally distributed throughout the truth table. This idea was first introduced by Webster and Tavares in [21]. Given $f(\mathbf{x}) \in \mathcal{B}_n$ and an input $\mathbf{x} = (x_1x_2, \ldots, x_n)$, if we select an x_k in $\mathbf{x}$ with $1 \leq k \leq n$, then we can envision the domain $\mathbb{F}_2^n$ as two equivalence classes, $A = \{(x_1, \ldots, x_n)|x_k = 0\}$ and $B = \{(x_1, \ldots, x_n)|x_k = 1\}$. We note that there are 2^{n-1} unique pairs $(\mathbf{x}, \mathbf{y})$ with $\mathbf{x} \in A$ and $\mathbf{y} \in B$ such that $x_i = y_i$ with $i = 1, 2, \ldots n$ except for when $i = k$. Without loss of generality, assume $x_k = 0$. As x_k changes from 0 to 1, some pairs have the same function values (are not affected by the change), and the others have their function values changed from 0 to 1 or 1 to 0. The Boolean function f satisfies the SAC, if exactly half of the pairs change their function values for all k.

Example 2.3.8. [4, p. 25] In Table 2.9, if we fix $x_2 = 0$, we have $f(000) = 1$, $f(001) = 1$, $f(100) = 0$, and $f(101) = 1$. When x_2 becomes 1, we have $f(010) = 1$, $f(011) = 0$, $f(110) = 1$, and $f(111) = 1$. We observe that as x_2 changes from 0 to 1, $f(0x_20)$ and $f(1x_21)$ do not change, but $f(0x_21)$ and $f(1x_20)$ change. We can check x_1 and x_2 in a similar manner and observe the same result. Therefore, f satisfies the SAC.

The next lemma is a well-known equivalent statement to the definition of the SAC.

$\mathbf{x}$	000	001	010	011	100	101	110	111
$f(\mathbf{x})$	1	1	1	0	0	1	1	1

Table 2.9: A 3-variable Function Which Satisfies the SAC

Lemma 2.3.9. [21] *A Boolean function f satisfies the SAC if and only if $C_{\hat{f}}(\mathbf{w}) = 0$ for all $wt(\mathbf{w}) = 1$ where $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ and $1 \leq i \leq n$.*

Using Lemma 2.3.9, we can develop a computational tool to verify if a Boolean function satisfies the SAC.

2.3.4.2. Propagation Criteria

The concept of the propagation criterion generalizes the SAC. Preneel et al. [22] first introduced this idea.

Definition 2.3.10. [4, p. 38] A Boolean function f satisfies the propagation criterion of degree k or $PC(k)$ if changing the value of any i elements of the input vector with $1 \leq i \leq k \leq n$ changes exactly the half of the function values of the affected vectors.

We can extend Lemma 2.3.9 to cover the $PC(k)$ functions.

Lemma 2.3.11. *A Boolean function f satisfies PC(k) if and only if $C_{\hat{f}}(\mathbf{w}) = 0$ for all $wt(\mathbf{w}) = m$ where $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ and $1 \leq m \leq k$.*

2.3.5. Global Avalanche Criterion (GAC)

In [9], Zhang and Zheng first introduced the concept of GAC. They noted that the functions with SAC provide some level of security, but the SAC is only "local" and does not cover all possible linear structures in a Boolean function. $PC(k)$ on the other hand covers all possibilities. It seems that a large k implies better security. However, when k is even and $k = n$, the function is a bent function. Despite the highest nonlinearity, a bent function is not balanced. To address these issues, they introduced GAC, in which we measure the avalanche effects throughout all possible n-variable Boolean vectors using the two measures below.

Definition 2.3.12. [9] Given a Boolean function $f(\mathbf{x})$, the *sum-of-squares indicator* for the avalanche characteristic of $f(\mathbf{x})$ is

$$\sigma_f = \sum_{\alpha \in \mathbb{F}_2^n} C_{\hat{f}}^2(\alpha),$$

and the *absolute indicator* for the characteristic is

$$\triangle_f = \max_{\alpha \in \mathbb{F}_2^n} \left| C_{\hat{f}}(\alpha) \right|,$$

where $C_{\hat{f}}(\alpha) = \sum_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \alpha)}$.

Some cryptographic properties conflict with one another. In this case we see three conflicting properties, namely balance, nonlinearity, and propagation criteria. The GAC provides us with two general measures that we can minimize.

2.3.6. Correlation Immunity and Resilience

Given some Boolean function values $f(\mathbf{x})$, an attacker may guess the relationship between the elements of input, x_i of $\mathbf{x} = (x_1, x_2, \ldots x_n)$ and $f(\mathbf{x})$. Therefore, we want to engineer a principle into our function to deal with this kind of situation. Siegenthaler [23] first conceived the notion of *correlation immunity* to address this issue.

Definition 2.3.13. [4, p. 49] Let $x_{c1}, x_{c2}, \ldots x_{ci}$ of $\mathbf{x} = (x_1, x_2, \ldots x_n)$ be any i variables with $i \leq k$ of input $\mathbf{x}$. A Boolean function $f(\mathbf{x}) \in \mathcal{B}_n$ has *correlation immunity* of order k, denoted by $CI(k)$, if given $f(\mathbf{x})$, the probability of $x_{c1}, x_{c2}, \ldots x_{ci}$ being certain value is 2^{-i}. In other words, $f(\mathbf{x})$ is statistically independent with respect to any subset of k variables. In particular, $f(\mathbf{x})$ is called a *resilient* function of order k if it is $CI(k)$ and balanced.

Example 2.3.14. The Boolean function in the Table 2.10 has $CI(1)$. For example, if $f(\mathbf{x}) = 0$ and $x_i = x_1$, we can compute the conditional probability with $x_i = 0$,

$$
\begin{aligned}
Pr(x_1 = 0 | f(\mathbf{x}) = 0) &= \frac{Pr(x_1 = 0 \cap f(\mathbf{x}) = 0)}{Pr(f(\mathbf{x}) = 0)} \\
&= \frac{3/8}{6/8} \\
&= \frac{1}{2}
\end{aligned}
$$

the conditional probability with $x_i = 1$,

$$
\begin{aligned}
Pr(x_1 = 1 | f(\mathbf{x}) = 0) &= \frac{Pr(x_1 = 1 \cap f(\mathbf{x}) = 0)}{Pr(f(\mathbf{x}) = 0)} \\
&= \frac{3/8}{6/8} \\
&= \frac{1}{2}.
\end{aligned}
$$

The same procedures can check for $x_i = x_2$, x_3 to conclude that the function has $CI(1)$. However, we observe that

$$
\begin{aligned}
Pr(x_1 = 1, x_2 = 1 | f(\mathbf{x}) = 0) &= \frac{Pr(x_1 = 1 \cap x_2 = 1 \cap f(\mathbf{x}) = 0)}{Pr(f(\mathbf{x}) = 0)} \\
&= \frac{1/8}{6/8} \\
&= \frac{1}{6} \\
&\neq \frac{1}{4}.
\end{aligned}
$$

Therefore, $f(\mathbf{x})$ does not have $CI(2)$.

$\mathbf{x}$	000	001	010	011	100	101	110	111
$f(\mathbf{x})$	0	0	0	1	1	0	0	0

Table 2.10: A three-variable function with CI(1)

There is an efficient way to verify CI using the Walsh-Hadamard transform.

Lemma 2.3.15. [4, p. 50] *Let $f \in \mathcal{B}_n$. $CI(f) = k$ with $1 \leq k \leq n$ if and only if*

$$W_f(\mathbf{w}) = \sum_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{f(\mathbf{x}) \oplus \mathbf{w} \cdot \mathbf{x}} = 0$$

for all $\mathbf{w}$ *where* $1 \leq wt(\mathbf{w}) \leq k$.

2.3.7. Algebraic Immunity

For decades, linearization and some of its variations have been used to attack a stream cipher employing a Boolean function. They typically use Gaussian elimination as a core algorithm. By choosing a Boolean function with a high degree, we can substantially increase the computing resources needed to carry out an attack, which renders linearization useless as a practical technique to solve a stream cipher. However, a new class of attack

was introduced in 2003. It was shown that if a stream cipher employs a Boolean function f or $f \oplus 1$ with a low-degree function such that $fg = 0$ or $(f \oplus 1)g = 0$, the cipher can be methodically solved by the algebraic attack discussed in [24] and [25].

Definition 2.3.16. For any $f \in \mathcal{B}_n$, a nonzero function $g \in \mathcal{B}_n$ is called an *annihilator* of f if $fg = \mathbf{0}$, and the *algebraic immunity* of f, denoted by $AI(f)$, is the minimum value of d such that f or $f \oplus 1$ admits an annihilator of degree d [26].

The following two cases are algebraic attack possibilities [24].

Case 1: Assume that there exists a function g of low algebraic degree such that $fg = h$, where h is a nontrivial function with low algebraic degree.

Case 2: Assume that there exists a function g of low algebraic degree such that $fg = 0$. In 2003, Courtois and Meier showed that the algebraic immunity of an n variable Boolean function is bounded above by $\lceil \frac{n}{2} \rceil$.

Remark 2.3.17. [27] While algebraic immunity is an important cryptographic property, it is not enough to resist fast algebraic attacks, a more efficient form of algebraic attacks. If we can find g of low degree and h of algebraic degree not much larger than $n/2$, such that $fg = h$, then f is susceptible to fast algebraic attacks [24], [28].

2.3.8. Normality

The normality was first discussed by Dobbertin while examining bent functions in [29]. Since the number of variables in a bent function is even, the initial focus was on the even variable functions, which are invariant with respect to the vectors in a flat. Dobbertin called a Boolean function of even variables "normal" if it is invariant on a flat of the dimension $\frac{n}{2}$. Later this concept was generalized for odd variable functions invariant in a flat of dimension $\lceil \frac{n}{2} \rceil$. Dobbertin conjectured that all bent functions are normal. However, some non-normal bent functions were discovered by Canteaut el al. [30], and the notion of normality became an independent measure for general Boolean functions. Later, it was shown that there are very few normal functions, and the definition below was established by Carlet [31].

Definition 2.3.18. A Boolean function $f \in \mathcal{B}_n$ is called k-*normal* if there exist a k dimensional flat G such that f is constant. We denote such condition as $f\ |_G = 0$ or 1. If $k = \lceil \frac{n}{2} \rceil$, f is simply called a *normal* function.

General information on the normality can be found in [32].

2.4. TRADEOFFS BETWEEN CRYPTOGRAPHIC PROPERTIES

Unfortunately, composing or finding good cryptographic Boolean functions has a few obstacles, since there are some cryptographic properties that we cannot optimize simultaneously. We present common dilemmas among cryptographic properties with the relevant theorems.

2.4.1. Correlation Immunity and Degree

In 1984, Siegenthaler [23] showed that there is a necessary tradeoff between achieving high-degree and high-correlation immunity.

Theorem 2.4.1. [23, Theorem 1] *If a Boolean function f is CI(k), then the degree of f is at most $n - k$. If f is CI(k) with $k < n - 1$ and balanced, then the degree of f is at most $n - k - 1$.*

2.4.2. Correlation Immunity and Nonlinearity

Theorem 2.4.2 illustrates the tradeoff between correlation immunity and nonlinearity of Boolean functions.

Theorem 2.4.2. [33] *If a Boolean function f is CI(k) with $k \leq n - 2$,*

$$nl(f) \leq 2^{n-1} - 2^{k+1}.$$

We can combine Theorems 2.4.1 and 2.4.2 and obtain the following theorems.

Theorem 2.4.3. [4, p. 71] *If f is balanced and CI(k) with $k \leq n - 2$, then equality is possible in Theorem 2.4.2 only if f has its maximum possible degree $n - k - 1$.*

If $deg(f) < n - k - 1$*, then*

$$nl(f) \leq 2^{n-1} - 2^{k+2}.$$

The following theorem by Carlet improves Theorem 2.4.3 to incorporate the degree of the function in the upperbound [4, p. 72].

Theorem 2.4.4. [34] *If a balanced Boolean function* f *with degree* d *is CI(*k*) with* $k \leq n - 2$*, then*

$$nl(f) \leq 2^{n-1} - 2^{k+1+\lfloor (n-k-2)/d \rfloor}.$$

2.4.3. Algebraic Immunity and Nonlinearity

The following theorem describes the limit (commonly called "Lobanov's bound"). The theorem implies that we can increase the algebraic immunity of a function along with the nonlinearity, but at the expense of decreasing the correlation immunity due to Theorem 2.4.2.

Theorem 2.4.5. [35] *If* $f \in \mathcal{B}_n$ *has algebraic immunity* k*,*

$$nl(f) \geq 2 \sum_{i=0}^{k-2} \binom{n-1}{i}.$$

3. AFFINE EQUIVALENCE OF MONOMIAL ROTATION-SYMMETRIC BOOLEAN FUNCTIONS

3.1. INTRODUCTION

In this chapter, we study the affine equivalence of monomial rotation-symmetric (MRS) Boolean functions. A general affine equivalence problem for Boolean functions is a complete partitioning of the n-variable Boolean function space based on an affine equivalence relation. A greedy algorithm for affine equivalence verification requires checking all elements of $GL_n(\mathbb{F}_2)$, and has computational complexity $O(2^{n^2})$. This implies that if $n \geq 7$, the problem becomes quite a challenge for current computing platforms. The first notable effort to solve an affine equivalence problem is found in [36], published in 1964. Berlekamp and Welch [37] in 1972 found all equivalence classes for all five variable Boolean functions. In 1991, Maiorana [38] computed $150,357$ equivalence classes of six variable Boolean functions. Due to its complexity and size, affine equivalence still remains a tough problem to deal with, especially for a general solution, which addresses any $n \in \mathbb{N}$. Besides the pure mathematical perspective, an affine equivalence can be applied to cryptanalysis and cryptographic engineering. For example, differential and linear cryptanalyses are two major techniques to solve the S-boxes of block ciphers. If an S-box is vulnerable to differential or linear cryptanalysis, so are the S-boxes realizing affine equivalence functions. This fact simplifies the tasks of cryptanalysts, since they just need to choose and analyze an (easy) representative of an equivalence class. On the other hand, the cryptographic engineers may take advantage of affine equivalent S-boxes of a S-box that is strongly resistant to these attacks, since affine transformations have small delays and preserve much of the cryptographic properties of the original function.

A rotation-symmetric Boolean function (RSBF) is invariant under the rotation or circular shift of a input. For example, if $f \in \mathcal{B}_3$ is rotation symmetric, then $f(001) = f(010) = f(100)$, $f(011) = f(101) = f(110)$, and so on. Since an RSBF uses re-

n	Number of Classes
1	1
2	2
3	3
4	8
5	48
6	150,357

Table 3.1: Affine Equivalence Classes in $\mathcal{B}_n$

peated function values, it is relatively fast. However, despite being seemingly simple functions to evaluate, the class of RSBFs contain many functions richly endowed with good cryptographic properties. For example, the famous Patterson–Wiedemann function in $\mathcal{B}_{15}$ [39] that achieves nonlinearity 16276, which is strictly greater than the bent concatenation bound, $2^{15-1} - 2^{(15-1)/2} = 16256$ is rotation symmetric [4, p. 112]. Moreover, Kavut et al. [40], [41], [42] proved that there exist rotation-symmetric functions of nine variables with the nonlinearity 241 and 242, which is also strictly greater than the bent concatenation bound $2^{9-1} - 2^{(9-1)/2} = 240$ [4, p. 112]. Due to their speed and the prospect of being good cryptographic Boolean functions, RSBFs have received a lot of attention from cryptographic researchers. In [43], Filiol and Fontaine initially studied cryptographic properties of RSBFs (they used the term, "idempotent" function instead of RSBF), mainly focusing on nonlinearity [4, p. 112]. Later, the nonlinearity and correlation immunity of RSBFs were studied thoroughly in [44], [45], [46], [47], and [48]. The RSBF's speed and potential to have good cryptographic properties make them suitable for such an application as hashing algorithms. Pieprzyk and Qu studied the use of RSBFs in a hashing algorithm in [3]. We note the papers [49] and [50] dealing with algebraic immunity of RSBF. The class of RSBFs are interesting to apply the notion of affine equivalence into, as the function space is much smaller ($\approx 2^{\frac{2^n}{n}}$) than the total space of Boolean functions (2^{2^n}), and the set contains functions with very good cryptographic properties. It has been experimentally demonstrated that there are RSBFs that are simultaneously good in terms of balancedness, nonlinearity, correlation immunity, algebraic degree, and algebraic immunity.

There has been consistent effort to investigate the affine equivalence of RSBFs. Some recent efforts include [51], [52], [53], [54], and [55]. In this chapter, we focus on a type of affine equivalence named "S-equivalence" applied to monomial rotation-symmetric (MRS) functions. The material in this chapter is based on Chung and Stanica [56].

3.2. AFFINE EQUIVALENCE OF BOOLEAN FUNCTIONS

Definition 3.2.1. We say that $f, g \in \mathcal{B}_n$ are *affine equivalent* if there exists an $n \times n$ invertible matrix A over the finite field $\mathbb{F}_2$, the vectors $\mathbf{b}, \mathbf{c} \in \mathbb{F}_2^n$ and $d \in \mathbb{F}_2$ such that $g(\mathbf{x}) = f(\mathbf{x}A \oplus \mathbf{b}) \oplus \mathbf{c} \cdot \mathbf{x} \oplus d$.

Some researchers prefer a simplified version of equivalence where $\mathbf{c} = \mathbf{0}$ and $d = 0$.

Definition 3.2.2. [55] We say that two Boolean functions $f(\mathbf{x})$ and $g(\mathbf{x})$ in $\mathcal{B}_n$ are *equivalent under an affine transformation* if $g(\mathbf{x}) = f(\mathbf{x}A \oplus \mathbf{b})$, where A is an $n \times n$ nonsingular matrix over the finite field $\mathbb{F}_2$ and $\mathbf{b}$ is an n-dimensional vector over $\mathbb{F}_2$. We say $f(\mathbf{x}A \oplus \mathbf{b})$ is a *nonsingular affine transformation* of $f(\mathbf{x})$.

In this thesis, we focus on a type of affine equivalence where $b = 0$, $c = 0$, $d = 0$, and A is permutation matrix. We will define this notion called "S-equivalence" in a later section.

Example 3.2.3. Consider the following five variable Boolean functions,

$$
\begin{aligned}
f &= x_1x_2 \oplus x_3x_4x_5 \\
f_1 &= x_1x_2 \oplus x_3x_4x_5 \oplus x_1 \oplus x_3 \\
f_2 &= x_1x_2 \oplus x_3x_4x_5 \oplus x_2 \oplus x_3 \oplus x_5 \oplus 1 \\
f_3 &= x_3x_4 \oplus x_1x_2x_5 \oplus x_1 \oplus x_3 \oplus 1 \\
f_4 &= (x_4 \oplus 1)x_3 \oplus x_1x_2(x_5 \oplus 1) \oplus x_1 \oplus x_3 \oplus 1 \\
&= x_3x_4 \oplus x_1x_2x_5 \oplus x_1x_2 \oplus x_1 \oplus 1
\end{aligned}
$$

We see that $f_1 = f \oplus \mathbf{c} \cdot \mathbf{x}$, where $\mathbf{c} = (1, 0, 1, 0, 0)$. $f_2 = f \oplus \mathbf{c} \cdot \mathbf{x} \oplus d$, where $\mathbf{c} = (0, 1, 1, 0, 1)$ and $d = 1$. $f_3 = f(\mathbf{x}A) \oplus \mathbf{c} \cdot \mathbf{x} \oplus d$, where

$$
A = \begin{pmatrix}
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0
\end{pmatrix},
$$

$\mathbf{c} = (1, 0, 1, 0, 0)$, and $d = 1$. $f_4 = f(\mathbf{x}A \oplus \mathbf{b}) \oplus \mathbf{c} \cdot \mathbf{x} \oplus d$, where A, $\mathbf{c}$, and d are same as f_3 with $\mathbf{b} = (1, 0, 0, 1, 0)$.

Essentially, a permutation transformation rearranges the order of input, which preserves the Hamming weight of the truth table. Clearly, if f and g are equivalent under

affine transformation, then $wt(f) = wt(g)$ and $nl(f) = nl(g)$. However, the sufficiency only holds for quadratic Boolean functions.

Theorem 3.2.4. [56] *Two quadratic functions f and g in $\mathcal{B}_n$ are equivalent under affine transformation if and only if $wt(f) = wt(g)$ and $nl(f) = nl(g)$.*

Unfortunately, the result cannot be extended to higher degrees. In S-equivalence, we obtain a similar result for degrees ≥ 2. If two functions f and g in $\mathcal{B}_n$ are S-equivalent, then $wt(f) = wt(g)$ and $nl(f) = nl(g)$. The converse of the statement does not hold. We can still use the result to show non-equivalence in many cases.

3.3. ROTATION-SYMMETRIC BOOLEAN FUNCTIONS

Definition 3.3.1. Let $x_i \in \mathbb{F}_2$ for $1 \leq i \leq n$. For $1 \leq k \leq n$, we define the permutation ρ_n^k on $(x_1, x_2, \ldots, x_n) \in \mathbb{F}_2^n$ such that

$$\rho_n^k((x_1, x_2, \ldots, x_{n-1}, x_n)) = (\rho_n^k(x_1), \rho_n^k(x_2), \ldots, \rho_n^k(x_{n-1}), \rho_n^k(x_n)),$$

where

$$\rho_n^k(x_i) = x_{i+k} \text{ if } i + k \leq n$$

and

$$\rho_n^k(x_i) = x_{i+k-n} \text{ if } i + k > n.$$

Hence, ρ_n^k acts as k-cyclic rotation on an n-bit vector.

Based on the permutation in Definition 3.3.1, we define the RSBF.

Definition 3.3.2. A Boolean function f is called *rotation symmetric* if, for each vector $(x_1, \ldots, x_n)$ in $\mathbb{F}_2^n$,

$$f(\rho_n^k(x_1, \ldots, x_n)) = f(x_1, \ldots, x_n), \text{ for } 1 \leq k \leq n.$$

Definition 3.3.2 implies that the rotation-symmetric Boolean functions (RSBFs) are invariant under a cyclic rotation of input vectors. Clearly, the input vectors in a rotation class are in a equivalence relation. Therefore, the inputs of a rotation-symmetric Boolean function can be divided into partitions so that each partition consists of all cyclic shifts of one input. A partition is generated by say $G_n(x_1, x_2, \ldots, x_n) = \{\rho_n^k(x_1, x_2, \ldots, x_n) | 1 \leq k \leq n\}$, and we denote the number of such partitions g_n. By the product rule of combinatorics, the number of n-variable RSBFs is 2^{g_n}. Let $\phi(k)$ be Euler's *phi*-function. Then, from Burnside's lemma [48],

$$g_n = \frac{1}{n} \sum_{k|n} \phi(k) 2^{\frac{n}{k}}.$$

Let $g_{n,w}$ denote the number of partitions with w, the common weight of the vectors in partition. The papers [45], [47], and [48] address the formula on how to calculate $g_{n,w}$ for arbitrary n and w. It is also noteworthy that Zhang and Deng [57] corrected the enumeration of $G_n(x_1, x_2, \ldots, x_n)$ such that $|G_n(x_1, x_2, \ldots, x_n)| = n$ in [48] and generalized the enumeration for $|G_n(x_1, x_2, \ldots, x_n)| = r$ where $r \mid n$.

Definition 3.3.3. Let

$$G_n(x_1, \ldots, x_n) = \{\rho_n^k(x_1, \ldots, x_n), \text{ for } 1 \leq k \leq n\},$$

be the orbit of $(x_1, \ldots, x_n)$ under the action of ρ_n^k, $1 \leq k \leq n$. It is clear that $G_n(x_1, \ldots, x_n)$ generates a partition in the set $\mathbb{F}_2^n$. A rotation-symmetric function $f(x_1, \ldots, x_n)$ can be

written (for short) as

$$a_0 \oplus a_1 x_1 \oplus \sum a_{1j} x_1 x_j \oplus \cdots \oplus a_{12\ldots n} x_1 x_2 \ldots x_n \, (SANF),$$

where the coefficients $a_0, a_1, a_{1j}, \ldots, a_{12\ldots n} \in \{0, 1\}$, and the existence of a representative term $x_1 x_{i_2} \ldots x_{i_l}$ implies the existence of all the terms from $G_n(x_1 x_{i_2} \ldots x_{i_l})$ in the ANF. We call this representation of f the *short algebraic normal form* (SANF) of f.

Remark 3.3.4. We note that the SANF is not unique, since one can choose any representative in $G_n(x_1 x_{i_2} \ldots x_{i_l})$.

Example 3.3.5. 5-variable RSBFs f and g are shown in ANF and SANF below.

$$\begin{aligned} f(\mathbf{x}) &= x_1 (SANF) \\ &= x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \end{aligned}$$

$$\begin{aligned} g(\mathbf{x}) &= x_1 \oplus x_1 x_2 x_5 (SANF) \\ &= x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_1 x_2 x_5 \oplus \cdots \oplus x_5 x_1 x_4 \end{aligned}$$

If the SANF of a RSBF contains only one term, we call such a function a *monomial rotation-symmetric* (MRS) function. A simple number theoretic deduction gives us that the ANF of a monomial rotation-symmetric function contains a divisor of n number of terms. If that divisor is in fact n, we call the function a *full-cycle MRS*, otherwise, a *short-cycle MRS*.

Example 3.3.6. 6-variable RSBF $f(\mathbf{x}) = x_1x_2(SANF)$ is a full-cycle MRS function, and $g(\mathbf{x}) = x_1x_4(SANF)$ are short-cycle MRS function, as shown below.

$$f(x) = x_1x_2 \oplus x_2x_3 \oplus x_3x_4 \oplus x_4x_5 \oplus x_5x_6 \oplus x_6x_1$$

$$g(x) = x_1x_4 \oplus x_2x_5 \oplus x_3x_6$$

3.4. CIRCULANT MATRICES

One of the interesting matrices in linear algebra is a Toeplitz matrix. An $n \times n$ Toeplitz matrix $A = \{a_{ij}\}$ has a form

$$A = \begin{pmatrix} a_1 & a_2 & a_3 & \cdots & \cdots & a_n \\ a_{n+1} & a_1 & a_2 & \ddots & & \vdots \\ a_{n+2} & a_{n+1} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_2 & a_3 \\ \vdots & & \ddots & a_{n+1} & a_1 & a_2 \\ a_{2n-1} & \cdots & \cdots & a_{n+2} & a_{n+1} & a_1 \end{pmatrix}.$$

Toeplitz matrices have various engineering applications and have been widely studied. A circulant matrix is a special type of Toeplitz matrix where $a_2 = a_{2n-1}$, $a_3 = a_{2n-2}$, ... , and $a_n = a_{n+1}$. We apply the principles found in the structure of a circulant matrix extensively in our new findings. To be precise, we use the following definition for a circulant matrix.

Definition 3.4.1. An $n \times n$ matrix C is circulant, denoted by $C(c_1, c_2, \ldots, c_n)$, if all its rows are successive circular permutations of the first row, that is,

$$C = \begin{pmatrix} c_1 & c_2 & c_3 & \cdots & \cdots & c_n \\ c_n & c_1 & c_2 & \ddots & & \vdots \\ c_{n-1} & c_n & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & c_2 & c_3 \\ \vdots & & \ddots & c_n & c_1 & c_2 \\ c_2 & \cdots & \cdots & c_{n-1} & c_n & c_1 \end{pmatrix},$$

where $c_i \in \mathbb{F}$ for $\mathbb{F}$ is a field, and $i \in \{1, 2, \ldots, n\}$.

We denote the set of all circulant matrices as $\mathcal{C}$ and the set of all $n \times n$ circulant matrices as $\mathcal{C}_n$.

We define the *generating polynomial* F of a circulant matrix $C(c_1, \ldots, c_n)$ by

$$F(\mathbf{x}) = c_1 + c_2 z + \cdots + c_n z^{n-1}.$$

It is clear that the circulant matrices are closed under matrix addition. That is, for any two circulant matrices A and B, $A+B$ is circulant as well. Additionally, $A+B = B+A$ and the associative property holds. Therefore, $\mathcal{C}_n$ forms an abelian group. We proceed to prove another interesting fact about circulant matrices. We also observe that the transpose of a circulant matrix $C = C(c_1, c_2, \ldots, c_n)$, denoted by C^T, is $C(c_1, c_n, c_{n-1}, \ldots, c_2)$

Proposition 3.4.2. [56] *An $n \times n$ matrix $C = \{c_{ij}\}$ is circulant if and only if $c_{ij} = c_{uv}$ whenever $j - i \equiv u - v \pmod{n}$.*

There exists a way to express a circulant matrix as a linear combination of a basis of matrices. Let G be the $n \times n$ binary circulant matrix $G = C(0, 1, 0, \ldots, 0)$, which is

$$G = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ \vdots & \vdots & & \ddots & 1 & 0 \\ 0 & 0 & \ddots & & 0 & 1 \\ 1 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix}.$$

The following lemma shows that the power of G, G^j, where $1 \leq j \leq n$, form a basis for the commutative algebra $\mathcal{C}_n$.

Lemma 3.4.3. [58, p. 68] *Let $A \in \mathcal{C}_n$ and $A = C(a_1, a_2, \ldots, a_n)$. Then*

$$A = \sum_{i=1}^{n} a_i G^{i-1} = \sum_{i \in \Delta(A)} G^{i-1},$$

where $\Delta(A) = \{i | \, a_i = 1\} \subseteq \{1, 2, \ldots, n\}$.

It is well-known that the circulant matrices in $\mathbb{C}$ commute in multiplication [58, p. 68]. Since some matrix properties in $\mathbb{C}$ may not hold in $\mathbb{F}_2$, we verify the commutativity.

Lemma 3.4.4. [56] *Let $A = C(a_1, a_2, \ldots, a_n)$ and $B = C(b_1, b_2, \ldots, b_n)$ be two elements of $\mathcal{C}_n$ with a_i, $b_i \in \mathbb{F}_2$ for $1 \leq i, \, j \leq n$. Then,*

$$
\begin{aligned}
AB &= BA \\
&= C\left(\sum_{\substack{i,j=1 \\ i+j\equiv 2 \pmod n}}^{n} a_i b_j, \sum_{\substack{i,j=1 \\ i+j\equiv 3 \pmod n}}^{n} a_i b_j, \ldots, \sum_{\substack{i,j=1 \\ i+j\equiv 1 \pmod n}}^{n} a_i b_j\right) \\
&= C\left(\sum_{\substack{i\in\Delta(A), j\in\Delta(B) \\ i+j\equiv 2 \pmod n}} a_i b_j, \sum_{\substack{i\in\Delta(A), j\in\Delta(B) \\ i+j\equiv 3 \pmod n}} a_i b_j, \ldots, \sum_{\substack{i\in\Delta(A), j\in\Delta(B) \\ i+j\equiv 1 \pmod n}} a_i b_j\right)
\end{aligned}
$$

where $\Delta(A) = \{i |\, a_i = 1\} \subseteq \{1, 2, \ldots, n\}$ *(ordered tuple).*

Proof. Let

$$
A = \begin{pmatrix}
a_1 & a_2 & a_3 & \cdots & \cdots & a_n \\
a_n & a_1 & a_2 & \ddots & & \vdots \\
a_{n-1} & a_n & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & a_2 & a_3 \\
\vdots & & \ddots & a_n & a_1 & a_2 \\
a_2 & \cdots & \cdots & a_{n-1} & a_n & a_1
\end{pmatrix}, \text{ and } B = \begin{pmatrix}
b_1 & b_2 & b_3 & \cdots & \cdots & b_n \\
b_n & b_1 & b_2 & \ddots & & \vdots \\
b_{n-1} & b_n & \ddots & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & b_2 & b_3 \\
\vdots & & \ddots & b_n & b_1 & b_2 \\
b_2 & \cdots & \cdots & b_{n-1} & b_n & b_1
\end{pmatrix}.
$$

$$
\begin{aligned}
AB &= \begin{pmatrix}
\sum_{\substack{i+j\equiv 2 \pmod n}}^{i,j=1} a_ib_j & \sum_{\substack{i+j\equiv 3 \pmod n}}^{i,j=1} a_ib_j & \cdots & \cdots & \sum_{\substack{i+j\equiv 1 \pmod n}}^{i,j=1} a_ib_j \\
\sum_{\substack{i+j\equiv 1 \pmod n}}^{i,j=1} a_ib_j & & & \ddots & \vdots \\
\sum_{\substack{i+j\equiv n \pmod n}}^{i,j=1} a_ib_j & & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots & \\
\vdots & & \ddots & & \vdots \\
\sum_{\substack{i+j\equiv 3 \pmod n}}^{i,j=1} a_ib_j & \sum_{\substack{i+j\equiv 4 \pmod n}}^{i,j=1} a_ib_j & \cdots & & \sum_{\substack{i+j\equiv 2 \pmod n}}^{i,j=1} a_ib_j
\end{pmatrix} \\
&= C\left(\sum_{\substack{i+j\equiv 2 \pmod n}}^{i,j=1} a_ib_j, \sum_{\substack{i+j\equiv 3 \pmod n}}^{i,j=1} a_ib_j, \ldots, \sum_{\substack{i+j\equiv n \pmod n}}^{i,j=1} a_ib_j, \sum_{\substack{i+j\equiv 1 \pmod n}}^{i,j=1} a_ib_j \right)
\end{aligned}
$$

Since $a_i,\ b_j \in \mathbb{F}_2$,

$$
\begin{aligned}
&= C\left(\sum_{\substack{i\in\Delta(A),j\in\Delta(B) \\ i+j\equiv 2 \pmod n}} b_ia_j, \sum_{\substack{i\in\Delta(A),j\in\Delta(B) \\ i+j\equiv 3 \pmod n}} b_ia_j, \ldots, \sum_{\substack{i\in\Delta(A),j\in\Delta(B) \\ i+j\equiv 1 \pmod n}} b_ia_j \right) \\
&= BA.
\end{aligned}
$$

Therefore, the claim holds. □

Clearly, $\mathcal{C}_n$ has the associative property with respect to matrix multiplication. Therefore, $\mathcal{C}_n$ forms a commutative monoid. Since $\mathcal{C}_n$ is an abelian group, $\mathcal{C}_n$ forms a commutative algebra. We recall $A \in \mathcal{C}_n$ implies that $A^T \in \mathcal{C}_n$. Then, we have $A^TA = AA^T$ by Theorem 3.4.4. Therefore, $\mathcal{C}_n$ is normal.

Corollary 3.4.5. [56] *Let $A = C(a_1, a_2, \ldots, a_n)$ be a circulant matrix over $\mathbb{F}_2$. Then*

$$A^2 = C\left(\bigoplus_{\substack{i,j=1 \\ i+j\equiv 2 \pmod n}}^{n} a_i a_j, \bigoplus_{\substack{i,j=1 \\ i+j\equiv 3 \pmod n}}^{n} a_i a_j, \ldots, \bigoplus_{\substack{i,j=1 \\ i+j\equiv n \pmod n}}^{n} a_i a_j, \bigoplus_{\substack{i,j=1 \\ i+j\equiv 1 \pmod n}}^{n} a_i a_j\right)$$

$$= \begin{cases} C(a_1, a_{\lceil n/2\rceil+1}, a_2, a_{\lceil n/2\rceil+2}, \ldots, a_{\lceil n/2\rceil}) & \text{if } n \text{ is odd.} \\ \\ C(a_1 + a_{n/2+1}, 0, a_2 + a_{n/2+1}, 0, \ldots, 0) & \text{if } n \text{ is even.} \end{cases}$$

Proof. Let

$$A = C(a_1, a_2, \ldots, a_n)$$

$$= \begin{pmatrix} a_1 & a_2 & a_3 & \cdots & \cdots & a_n \\ a_n & a_1 & a_2 & \ddots & & \vdots \\ a_{n-1} & a_n & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_2 & a_3 \\ \vdots & & \ddots & a_n & a_1 & a_2 \\ a_2 & \cdots & \cdots & a_{n-1} & a_n & a_1 \end{pmatrix}.$$

By Lemma 3.4.4, we have

$$A^2 = C\left(\bigoplus_{\substack{i,j=1 \\ i+j\equiv 2 \pmod n}}^{n} a_i a_j, \bigoplus_{\substack{i,j=1 \\ i+j\equiv 3 \pmod n}}^{n} a_i a_j, \ldots, \bigoplus_{\substack{i,j=1 \\ i+j\equiv n \pmod n}}^{n} a_i a_j, \bigoplus_{\substack{i,j=1 \\ i+j\equiv 1 \pmod n}}^{n} a_i a_j\right).$$

If $n = 2k + 1$ for $k = 0, 1, 2, ...,$

$$\bigoplus_{\substack{i=1 \\ i+j\equiv 2 \pmod n}}^{n} a_i a_j = a_1a_1 \oplus a_2a_{2k+1} \oplus \cdots \oplus a_{k+1}a_{k+2} \oplus a_{k+2}a_{k+1} \oplus \cdots \oplus a_{2k+1}a_2$$

$$= a_1^2 \oplus 2a_{2k+1}a_2 \oplus \cdots \oplus 2a_k a_{k+3} \oplus 2a_{k+1}a_{k+2} = a_1$$

$$\bigoplus_{\substack{i=1 \\ i+j\equiv 3 \pmod n}}^{n} a_i a_j = a_1a_2 \oplus a_2a_1 \oplus a_3a_{2k+1} \oplus \cdots \oplus a_{k+2}^2 \oplus \cdots \oplus a_{2k+1}a_3$$

$$= a_{k+2}^2 \oplus 2a_1a_2 \oplus \cdots \oplus 2a_3a_{2k+1} = a_{k+2}$$

$$\bigoplus_{\substack{i=1 \\ i+j\equiv 4 \pmod n}}^{n} a_i a_j = a_1a_3 \oplus a_2a_2 \oplus a_3a_1 \oplus a_4a_{2k+1} \oplus \cdots \oplus a_{2k+1}a_4$$

$$= a_2^2 \oplus 2a_3a_1 \oplus \cdots \oplus 2a_{2k+1}a_4 = a_2$$

$$\vdots$$

$$\bigoplus_{\substack{i=1 \\ i+j\equiv 1 \pmod n}}^{n} a_i a_j = a_1a_{2k+1} \oplus a_2a_{2k} \oplus \cdots \oplus a_{k+1}^2 \oplus \cdots \oplus a_{2k}a_2 \oplus a_{2k+1}a_1$$

$$a_{k+1}^2 \oplus 2a_1a_{2k+1} \oplus \cdots \oplus 2a_{2k}a_2 = a_{k+1}.$$

Therefore,

$$A^2 = C\left(a_1, a_{k+2}, a_2, a_{k+3}, a_3, \ldots, a_k, a_{2k+1}, a_{k+1}\right)$$

$$= C(a_1, a_{\lceil n/2\rceil+1}, a_2, a_{\lceil n/2\rceil+2}, \ldots, a_{\lceil n/2\rceil}).$$

If $n = 2k$ for $k = 0, 1, 2, ...,$

$$\begin{aligned}
\bigoplus_{\substack{i=1 \\ i+j\equiv 2 \pmod n}}^{n} a_i a_j &= a_1a_1 \oplus a_2a_{2k} \oplus \cdots \oplus a_k a_{k+2} \oplus a_{k+1}a_{k+1} \oplus a_{k+2}a_k \oplus \cdots \oplus a_{2k}a_2 \\
&= a_1^2 \oplus a_{k+1}^2 \oplus 2a_2a_{2k} \oplus \cdots \oplus 2a_k a_{k+2} = a_1 \oplus a_{k+1} \\
\bigoplus_{\substack{i=1 \\ i+j\equiv 3 \pmod n}}^{n} a_i a_j &= a_1a_2 \oplus a_2a_1 \oplus a_3a_{2k} \oplus a_4a_{2k-1} \oplus \cdots \oplus a_{2k-1}a_4 \oplus a_{2k}a_3 \\
&= 2a_1a_2 \oplus \cdots \oplus 2a_{2k}a_3 = 0 \\
\bigoplus_{\substack{i=1 \\ i+j\equiv 4 \pmod n}}^{n} a_i a_j &= a_1a_3 \oplus a_2a_2 \oplus a_3a_1 \oplus a_4a_{2k} \oplus \cdots \oplus a_{k+2} \oplus \cdots \oplus a_{2k}a_4 \\
&= a_2^2 \oplus a_{k+2} \oplus 2a_3a_1 \oplus \cdots \oplus 2a_{2k}a_4 = a_2 \oplus a_{k+2} \\
&\vdots \\
\bigoplus_{\substack{i=1 \\ i+j\equiv 1 \pmod n}}^{n} a_i a_j &= a_1a_{2k} \oplus a_2a_{2k} \oplus \cdots \oplus a_{2k}a_2 \oplus a_{2k}a_1 \\
& \quad 2a_1a_{2k} \oplus \cdots \oplus 2a_{2k}a_2 = 0.
\end{aligned}$$

Therefore,

$$A^2 = C\left(a_1 + a_{k+1}, 0, a_2 \oplus a_{k+2}, 0, a_3 \oplus a_{k+3}, \ldots, a_k a_{2k}, 0\right)$$

$$= C(a_1 \oplus a_{n/2+1}, 0, a_2 \oplus a_{n/2+1}, 0, \ldots, 0)$$

□

An $n \times n$ *permutation* matrix P_σ is an $n \times n$ matrix obtained by applying a permutation $\sigma \in S_n$, where S_n is the symmetric group of the order n to the rows (or columns) of the identity matrix I_n.

Definition 3.4.6. We define a relation denoted by $\sim$ on $\mathcal{C}_n$ as follows. Let $A_1 = C(a_1, \ldots, a_n)$, $A_2 = C(b_1, \ldots, b_n)$. Then,

$$A_1 \sim A_2 \text{ if and only if } (a_1, \ldots, a_n) = \rho^k(b_1, \ldots, b_n).$$

Due to reflexivity, symmetry, and transitivity of the relation, the relation $\sim$ is an equivalence relation, which partitions $\mathcal{C}$ into equivalence classes. We denote the set of the equivalent classes as $\mathcal{C}/_\sim$. We further denote the equivalence class of $C(a_1, a_2, \ldots, a_n)$ by $C\langle a_1, a_2, \ldots, a_n\rangle$ or $\langle C(a_1, a_2, \ldots, a_n)\rangle$.

Lemma 3.4.7. [56] *Let M_1, $M_2 \in \mathcal{C}_n$, and let M_1^{-1} and M_2^{-1} exist. Then, M_1 and M_2 belong to the same equivalence class if and only if M_1^{-1} and M_2^{-1} also belong to the same equivalence class.*

Proof. We just prove the necessity; the sufficiency proof is similar. Let $M_1 = C(a_1, a_2, \ldots, a_n)$, $M_2 = C(b_1, b_2, \ldots, b_n)$ and $M_1^{-1} = C(\alpha_1, \alpha_2, \ldots, \alpha_n)$ and $M_2^{-1} = C(\beta_1, \beta_2 \ldots, \beta_n)$. It is sufficient to show that $M_2^{-1} \in C\left\langle \alpha_1, \alpha_2, \ldots, \alpha_n \right\rangle$. We know that

$$(b_1, b_2, \ldots, b_n) = \rho^k(a_1, a_2, \ldots, a_n)$$

for some k. Thus,

$$M_2 = P_k M_1$$

for some permutation matrix $P_k = C(\rho^k(1, 0, \ldots, 0))$. Therefore, by taking the inverse of the previous equation and Lemma 3.4.4,

$$\begin{aligned} M_1^{-1} &= M_2^{-1} P_k \\ &= P_k M_2^{-1}. \end{aligned}$$

Therefore, M_1^{-1} and M_2^{-1} belong to the same equivalence class. □

To conclude this section, we show that the equivalence classes of Definition 3.4.6 form a commutative monoid which contains a abelian group.

Theorem 3.4.8. [56] *The set $(\mathcal{C}/_\sim, \cdot)$ with the operation $\langle A\rangle \cdot \langle B\rangle := \langle AB\rangle$ is a commutative monoid. Moreover, the previous operation partitions the invertible circulant matrices $\mathcal{C}$ into equivalent classes, say $\mathcal{C}^*/_\sim$, and consequently, $(\mathcal{C}^*/_\sim, \cdot)$ becomes a group.*

Proof. First, we show that the operation is well defined. Let $A = C(a_1, \ldots, a_n) \sim A' = C(a'_1, \ldots, a'_n)$, $B = C(b_1, \ldots, b_n) \sim B' = C(b'_1, \ldots, b'_n)$. We need to show that $AB \sim A'B'$. By Lemma 3.4.4,

$$AB = C\left(\sum_{\substack{i,j=1 \\ i+j\equiv 2 \pmod n}}^{n} a_i b_j, \sum_{\substack{i,j=1 \\ i+j\equiv 3 \pmod n}}^{n} a_i b_j, \ldots, \sum_{\substack{i,j=1 \\ i+j\equiv 1 \pmod n}}^{n} a_i b_j\right)$$

$$A'B' = C\left(\sum_{\substack{i,j=1 \\ i+j\equiv 2 \pmod n}}^{n} a'_i b'_j, \sum_{\substack{i,j=1 \\ i+j\equiv 3 \pmod n}}^{n} a'_i b'_j, \ldots, \sum_{\substack{i,j=1 \\ i+j\equiv 1 \pmod n}}^{n} a'_i b'_j\right).$$

Let k and s be such that

$$\begin{aligned}\rho^k(a_1,\ldots,a_n) &= (a_{1+k \pmod n},\ldots,a_{n+k \pmod n})\\ &= (a'_1,\ldots,a'_n)\end{aligned}$$

and

$$\begin{aligned}\rho^s(b_1,\ldots,b_n) &= (b_{1+s \pmod n},\ldots,b_{n+s \pmod n})\\ &= (b'_1,\ldots,b'_n).\end{aligned}$$

Then, we have

$$\begin{aligned}A'B' &= C\left(\sum_{\substack{i,j=1\\ i+j\equiv 2 \pmod n}}^{n} a_{i+k \pmod n}b_{j+s \pmod n},\ldots,\sum_{\substack{i,j=1\\ i+j\equiv 1 \pmod n}}^{n} a_{i+k \pmod n}b_{j+s \pmod n}\right)\\ &= C\left(\sum_{\substack{i,j=1\\ i+j+k+s\equiv 2 \pmod n}}^{n} a_ib_j,\sum_{\substack{i,j=1\\ i+j+k+s\equiv 3 \pmod n}}^{n} a_ib_j,\ldots,\sum_{\substack{i,j=1\\ i+j+k+s\equiv 1 \pmod n}}^{n} a_ib_j\right)\\ &= C\left(\rho^{k+s}\left(\sum_{\substack{i,j=1\\ i+j\equiv 2 \pmod n}}^{n} a_ib_j,\sum_{\substack{i,j=1\\ i+j\equiv 3 \pmod n}}^{n} a_ib_j,\ldots,\sum_{\substack{i,j=1\\ i+j\equiv 1 \pmod n}}^{n} a_ib_j\right)\right).\end{aligned}$$

Therefore, we have

$$AB \sim A'B'.$$

It is immediate that the defined operation is associative, and the identity element is $C\langle 1,0,\ldots,0\rangle = \langle I_n\rangle$, the class of the identity matrix. The commutative property follows from Lemma 3.4.4. By Lemma 3.4.7, we can let $\langle M\rangle^{-1}$ be the equivalence class of all inverses of circulant matrices from $\langle M\rangle$. We have

$$\begin{aligned}\langle M\rangle \cdot \langle M\rangle^{-1} &= \langle M\rangle \cdot \langle M^{-1}\rangle \\ &= \langle I_n\rangle,\end{aligned}$$

and the lemma is proved. □

3.5. S-EQUIVALENCE OF MRS BOOLEAN FUNCTIONS

Definition 3.5.1. Let $f,\ g \in \mathcal{B}_n$be MRS functions. f and g are *S-equivalent*, denoted by $f \overset{s}{\sim} g$ if there exists a permutation matrix P such that

$$g(\mathbf{x}) = f(\mathbf{x}P).$$

Example 3.5.2. [56] Let $n = 7$, and the quartic MRS functions

$$\begin{aligned}f(\mathbf{x}) &= x_1x_2x_3x_4 \oplus x_2x_3x_4x_5 \oplus x_3x_4x_5x_6 \oplus x_4x_5x_6x_7 \\ &\quad \oplus x_5x_6x_7x_1 \oplus x_6x_7x_1x_2 \oplus x_7x_1x_2x_3,\end{aligned}$$

$$\begin{aligned}g(\mathbf{x}) &= x_1x_2x_4x_6 \oplus x_2x_3x_5x_7 \oplus x_3x_4x_6x_1 \oplus x_4x_5x_7x_2 \\ &\quad \oplus x_5x_6x_1x_3 \oplus x_6x_7x_2x_4 \oplus x_7x_1x_3x_5\end{aligned}$$

Using the permutation $\pi = (2,3,5)(4,7,6)$ expressed in product of disjoint cycles, we check that $f \circ \pi = g$.

We associate f to the following circulant matrix equivalence class

$$\begin{aligned} A_f &= C\langle \overset{1}{\underset{\downarrow}{1}}, 0, \ldots, \overset{j_2}{\underset{\downarrow}{1}}, 0, \ldots, 0, \overset{j_3}{\underset{\downarrow}{1}}, \ldots, 0, \overset{j_d}{\underset{\downarrow}{1}}, \ldots, 0 \rangle \\ &= \langle C(\overset{1}{\underset{\downarrow}{1}}, 0, \ldots, \overset{j_2}{\underset{\downarrow}{1}}, 0, \ldots, 0, \overset{j_3}{\underset{\downarrow}{1}}, \ldots, 0, \overset{j_d}{\underset{\downarrow}{1}}, \ldots, 0) \rangle, \end{aligned} \tag{3.1}$$

where the 1's appear in positions prompted by the indices of any monomial of ANF of f. We can illustrate $\Delta(f) = \Delta(\textit{any representative of } A_f)$. In general, for A_f as in Equation (3.1), then $\Delta(f) = [1, j_2, \ldots, j_d] = [2, j_2 + 1, \ldots, j_d + 1] = \cdots$. Also, the length of $\Delta(A)$ is denoted by $wt(\Delta(A))$, which is the weight of any row of A_f.

Example 3.5.3. [56] If $n = 5$ and $f(\mathbf{x}) = x_1x_2x_4 \oplus x_2x_3x_5 \oplus x_3x_4x_1 \oplus x_4x_5x_2 \oplus x_5x_1x_3$, then

$$A_f = \left\langle \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \right\rangle$$

$$\Delta(f) = [1,2,4] = [2,3,5] = [1,3,4] = [2,4,5] = [1,3,5].$$

Lemma 3.5.4. [56] *Let f be an MRS Boolean function, and F_i, $i = 1, 2$, be the generating polynomials for the circulant matrices $M_1 = C(a_1, a_2, \ldots, a_n)$, respectively, $M_2 = C(b_1, \ldots, b_n)$ in A_f, where $(b_1, \ldots, b_n) = \rho^k(a_1, \ldots, a_n)$, for some k. Then,* $\gcd(F_1(z),\ z^n - 1) = \gcd(F_2(z),\ z^n - 1)$.

Proof. Since $(b_1, b_2, \ldots, b_n) = \rho^k(a_1, a_2, \ldots, a_n)$, for some k, we use an inductive argument to prove the lemma. Let $k = 1$. Then, $(b_1, b_2, \ldots, b_n) = (a_n, a_0, \ldots, a_{n-2})$. Now, we need to show that

$$\gcd(F_1(z), z^n - 1) = \gcd(F_2(z), z^n - 1)$$

for

$$F_1(z) = a_1 + a_2 z + \cdots + a_n z^{n-1}$$

and

$$F_2(z) = a_n + a_1 z + \cdots + a_{n-1} z^{n-1}.$$

Certainly,

$$zF_1(z) - F_2(z) = a_n(z^n - 1). \tag{3.2}$$

Since multiplying z by $F_1(z)$ does not change $\gcd(F_1(z), z^n - 1)$,

$$\gcd(F_1(z), z^n - 1) \quad = \quad \gcd(zF_1(z), z^n - 1).$$

By Equation 3.2

$$\gcd(F_1(z), z^n - 1) \quad = \quad \gcd(a_n(z^n - 1) + F_2(z), z^n - 1).$$

By the Euclidean algorithm,

$$\gcd(F_1(z), z^n - 1) \quad = \quad \gcd(F_2(z), z^n - 1).$$

For the inductive step, assume it is true for $k = s$. Then, we try to show for $k = s + 1$. Let, $(b_1, b_2, \ldots, b_n) = (a_{n-s}, a_{n-s+1}, \ldots, a_{n-s-1})$. We need to show that

$$\gcd(F_1(z), z^n - 1) = \gcd(F_{s+1}(z), z^n - 1)$$

for

$$F_1(z) = a_1 + a_2 z + \cdots + a_n z^{n-1}$$

and

$$F_{s+1}(z) = a_{n-s} + a_{n-s+1} z + \cdots + a_{n-s-1} z^{n-1}.$$

Let

$$F_s(z) = a_{n-s+1} + a_{n-s+2} z + \cdots + a_{n-s} z^{n-1}.$$

Then,

$$zF_s(z) - F_{s+1}(z) = a_{n-s}(z^n - 1). \tag{3.3}$$

Since multiplying z by $F_s(z)$ does not change $\gcd(F_s(z), z^n - 1)$,

$$\gcd(F_s(z), z^n - 1) \quad = \quad \gcd(zF_s(z), z^n - 1).$$

By Equation 3.3

$$\gcd(F_s(z), z^n - 1) \quad = \quad \gcd(a_{n-s}(z^n - 1) + F_{s+1}(z), z^n - 1).$$

By the Euclidean algorithm,

$$\gcd(F_s(z), z^n - 1) \quad = \quad \gcd(F_{s+1}(z), z^n - 1).$$

By the induction hypothesis, we conclude that

$$\gcd(F_1(z), z^n - 1) = \gcd(F_{s+1}(z), z^n - 1),$$

which proves the lemma. □

We introduce the concept of a generalized inverse.

Definition 3.5.5. For a square matrix A, we call a matrix A^* of the same dimension a *generalized inverse* if

$$AA^*A = A.$$

We call a matrix $A^\dagger$ a *reflexive generalized* matrix if

$$AA^\dagger A = A$$

and

$$A^\dagger AA^\dagger = A^\dagger.$$

In addition, if both $AA^\dagger$ and $A^\dagger A$ are symmetric, then $A^\dagger$ is called a *(Moore–Penrose) pseudoinverse* of A. [59].

It is known that matrices over finite fields have at least one generalized inverse [60]. Also, if a pseudoinverse exists, it is unique [60]. However, it is not known if any of these generalized inverses of circulant matrices are circulant. Our next result deals with that problem, and, in the process, the first part generalizes the second, which was shown in [61, Theorem 2.2].

Theorem 3.5.6. [56] *Let $A = C(a_1, \ldots, a_n)$ be a circulant matrix over $\mathbb{F}_2$ of the generating polynomial $F = a_1 + a_2 z + \cdots + a_{n-1} z^n \in \mathbb{F}_2[z]$. Let $\gcd(F(z),\, z^n - 1) = D(z)$, $z^n - 1 = H(z) \cdot D(z)$, and assume that $\gcd(D(z), H(z)) = 1$. Then, the following statements hold:*

(i) The polynomial F is invertible modulo H. That is, there exists $F^*(z) = \sum_{j=1}^{n} \alpha_j z^{j-1}$ with $F(z) \cdot F^*(z) \equiv 1 \pmod{H(z)}$. Moreover, the circulant matrix A has a circulant generalized inverse, precisely, $A \cdot A^* \cdot A = A$, where $A^* = C(\alpha_1, \ldots, \alpha_n)$. Additionally, if $\gcd(F, z^n - 1) = \gcd(F^*, z^n - 1)$, then A^* is in fact the reflexive generalized inverse $A^\dagger$.

(ii) [61, Theorem 2.2] *If $\gcd(F, z^n - 1) = 1$, then the matrix A is invertible and its inverse is $A^{-1} = C(\alpha_1, \ldots, \alpha_n)$, where $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ is the unique solution of*

$$(\alpha_1, \alpha_2, \ldots, \alpha_n) \cdot A = (1, 0, \ldots, 0).$$

Moreover, if $F^*(z) = \alpha_1 + \alpha_2 z + \cdots + \alpha_n z^{n-1}$, then $F(z) \cdot F^*(z) \equiv 1 \bmod (z^n - 1)$.

Proof. The claim (ii) follows from (i). To show (i), let $n = 2^t m$ with m odd, and t an arbitrary integer. By [62, p.63 Theorem 2.42 (ii)], every irreducible factor of $z^n - 1$ over $\mathbb{F}_2$ appears at the power 2^t. Let $\Phi(z)$ be an arbitrary irreducible factor of $H(z) = (z^n - 1)/D(z)$. Since $\gcd(D(z), H(z)) = 1$, $\gcd(F(z), \Phi(z)) = 1$. Therefore, the class of $F(z)$ is invertible in the ring $\mathbb{F}_2[z]/\langle \Phi^{2^t} \rangle$. This implies that there exists $F_\Phi(z)^*$ with $F(z) \cdot F_\Phi(z)^* \equiv 1 \pmod{\Phi^{2^t}}$. Using the fact that $H(z) = \prod_{\Phi \ distinct} \Phi^{2^t}$ and applying the Chinese remainder theorem, we obtain that there exists F^* with $F(z) \cdot F^*(z) \equiv 1 \pmod{H(z)}$. Moreover, $F^*(z)$ is unique modulo $H(z)$.

To show the second claim of (i), we assume that $F \cdot F^* \equiv 1 \pmod{H}$, where $F^*(z) = \sum_{j=1}^{n} \alpha_j z^{j-1}$, and we will show that $AA^*A = A$, where $A^* = C(\alpha_1, \ldots, \alpha_n)$. Let R be the quotient ring $\mathbb{F}_2[z]/\langle H(z) \rangle$. Since D divides F and H divides $FF^* - 1$, then $z^n - 1 = HD$ divides $F(FF^* - 1)$ and so, we have the identity $F^2 F^* = F$ in $\mathbb{F}_2[z]/\langle z^n - 1 \rangle$. Multiplying out the polynomials F^2 and F^* and reducing modulo $z^n - 1$,

we obtain

$$\sum_{2i+k\equiv 3 \pmod n} a_i\alpha_k + \left(\sum_{2i+k\equiv 4 \pmod n} a_i\alpha_k\right) z + \cdots + \left(\sum_{2i+k\equiv 2 \pmod n} a_i\alpha_k\right) z^{n-1} = \sum_{\ell=1}^{n} a_\ell z^{\ell-1},$$

from which we infer that

$$C\left(\sum_{2i+k\equiv 3 \pmod n} a_i\alpha_k, \sum_{2i+k\equiv 4 \pmod n} a_i\alpha_k, \ldots, \sum_{2i+k\equiv 2 \pmod n} a_i\alpha_k\right) = C(a_1, a_2, \ldots, a_n).$$

That is $AA^*A = A$.

Using $\gcd(F(z), z^n - 1) = \gcd(F^*(z), z^n - 1)$, by a similar argument as before, we get that A is also a generalized inverse for A^*, that is, $A^*AA^* = A^*$, which shows the last claim of (i). □

As for the pseudoinverse, we observe that the transpose of a circulant matrix $A = C(a_1, a_2, \ldots, a_n)$ is $A^t = C(a_1, a_n, \ldots, a_2)$. Let $i' = (n + 2 - i) \mod n$, and $k' = (n + 2 - k) \mod n$. Then, we have

$$AA^* = C\left(\sum_{i+k\equiv 2 \pmod n} a_i\alpha_k, \sum_{i+k\equiv 3 \pmod n} a_i\alpha_k, \ldots, \sum_{i+k\equiv 1 \pmod n} a_i\alpha_k\right)$$

and

$$\begin{aligned}(AA^*)^t &= C\left(\sum_{i+k\equiv 2 \pmod n} a_{i'}\alpha_{k'}, \sum_{i+k\equiv 3 \pmod n} a_{i'}\alpha_{k'}, \ldots, \sum_{i+k\equiv 1 \pmod n} a_{i'}\alpha_{k'}\right) \\ &= C\left(\sum_{i'+k'\equiv 2 \pmod n} a_{i'}\alpha_{k'}, \sum_{i'+k'\equiv 1 \pmod n} a_{i'}\alpha_{k'}, \ldots, \sum_{i'+k'\equiv 3 \pmod n} a_{i'}\alpha_{k'}\right),\end{aligned}$$

which does not necessarily imply that $AA^* = (AA^*)^t$.

Remark 3.5.7. [56] It may be tempting to conjecture that every circulant matrix has a generalized inverse that is circulant. However, during a computer exercise, we noticed that the circulant matrix $C(1,0,0,1,0,0)$ does not have a circulant generalized inverse. We observe that $C(1,0,0,1,0,0)$ corresponds to $F(z) = 1 + z^3$ with $n = 6$. Since $z^6 - 1 = F(z)^2$,

$$H(z) = D(z) = F(z).$$

So, we have

$$gcd(D, H) \neq 1.$$

Therefore, Theorem 3.5.6 does not apply, and F has no inverse modulo F.

We mention another way to detect singularity or nonsingularity of the associated circulant matrix to an MRS. In [46], Stanica et al. found a characterization of Boolean functions whose associated circulant matrices are singular.

Proposition 3.5.8. [46] *Let f be a degree d MRS with associated $A_f = C\langle a_1, \ldots, a_n\rangle$ (assume that $a_1 = 1$). Let $\Delta(A_f) = [1, s_2, \ldots, s_d]$. Then, A_f is singular if and only if there is an n-th root of unity μ such that $1 + \mu^{s_2} + \cdots + \mu^{s_d} = 0$ (over $\mathbf{Z}_2$).*

Corollary 3.5.9. [46] *With the notation of the previous proposition, we have*

(i) *If* $wt(\Delta(A_f))$ *is even, then* A_f *is singular.*

(ii) *Let* p *be the least odd prime occurring in the factorization of* n*. Assume that* $\Delta(A_f) = [1, s_2, \ldots, s_d]$ *has odd weight* d *and* $s_d \leq p-2$*. Then* A_f *is nonsingular.*

We define the dual function with respect to a degree d MRS function f with invertible A_f. We consider the ordered set $\Delta(A_f^{-1}) = [j_1, j_2, \ldots, j_t]$ and define the MRS *dual function* f^* by

$$f^* = x_{j_1} x_{j_2} \cdots x_{j_t}(SANF).$$

Our next result gives an extension for the necessity part of Theorem 3.2.4.

Theorem 3.5.10. [56] *Let* f *and* g *be two MRS Boolean functions in* n*-variables. If* A_f *and* A_g *are invertible and* $f \overset{s}{\sim} g$ *(*f *and* g *are affine equivalent by a permutation in* S_n*), then* $wt(\Delta(f)) = wt(\Delta(g))$ *and* $wt(\Delta(f^*)) = wt(\Delta(g^*))$.

Proof. Since $f \overset{s}{\sim} g$, then there exists a permutation $\tau \in S_n$ with $A_{f\circ\tau} = A_g$. Clearly, f and g have the same degrees. Therefore, $wt(\Delta(f)) = wt(\Delta(g))$. Let the SANF of f be $x_1 x_{j_2} \cdots x_{j_d}$with $I = \{1, j_2, \ldots, j_d\}$. We set $A_f = \langle C(a_1, \ldots, a_n)\rangle$ such that $a_i = 1$ if $i \in I$, and 0 otherwise. Using the same steps, we also let $A_f^{-1} = \langle C(\alpha_1, \ldots, \alpha_n)\rangle$, $A_g = \langle C(b_1, \ldots, b_n)\rangle$, and $A_g^{-1} = \langle C(\beta_1, \ldots, \beta_n)\rangle$. Then we have

$$\langle C(b_1, \ldots, b_n)\rangle = \left\langle C(a_{\pi(1)}, a_{\pi(2)}, \ldots, a_{\pi(n)})\right\rangle,$$

since $A_g = A_{f\circ\tau}$, where $\pi = \tau^{-1}$. We introduce the notations $r_i(A)$ and $c_i(A)$ for the i-th row and the j-th column of a matrix A, respectively. Since the permutation τ preserves the rotation symmetry, there exists a permutation matrix such that every row of PA_g (not a circulant matrix any longer) is the permutation of the same indexed row of A_f. Then we have

$$r_i(PA_g) = \pi(r_i(A_f)).$$

By the hypothesis, there exists the inverse matrix

$$U = A_f^{-1} = \langle C(\alpha_1, \ldots, \alpha_n) \rangle .$$

Therefore, we have

$$r_i(A_f)U = r_i(I_n)$$

and

$$r_i(U)A_f = r_i(I_n).$$

Then, we can set

$$\begin{aligned} r_i(A_f) &= (a_{i,1}, a_{i,2}, \ldots, a_{i,n}) \\ &= (a_{n-i+2}, \ldots, a_{n-i+1}), \end{aligned}$$

which is the i-th shift of the first row of A_f. Let $\delta_{i,j}$ be the Kronecker delta function, that is, $\delta_{i,j} = 1$ if $i = j$, and $\delta_{i,j} = 0$ otherwise. Since π is a permutation, we can interpret the equation $A_f U = I_n$ in the following way:

$$a_{i,\pi(1)}u_{\pi(1),j} + \cdots + a_{i,\pi(n)}u_{\pi(n),j} = \delta_{i,j},\ 1 \leq i,\ j \leq n. \tag{3.4}$$

Let

$$U_{(\pi)} = \begin{pmatrix} u_{\pi(1),1} & \cdots & u_{\pi(n),n} \\ u_{\pi(1),1} & \cdots & u_{\pi(n),n} \\ \\ \cdots & \cdots & \cdots \\ \\ u_{\pi(1),1} & \cdots & u_{\pi(n),n} \end{pmatrix}.$$

Then, we have

$$U_{(\pi)} = P_\pi U$$

where P_π is the permutation matrix for π. Therefore, Equation (3.4) is simply $r_i(PA_g)c_j(U_{(\pi)}) = \delta_{i,j}$. Therefore,

$$PA_gU_{(\pi)} = I_n$$

and

$$U_{(\pi)}^{-1}PA_g = I_n.$$

Therefore,

$$r_1\left(U_{(\pi)}^{-1}P\right)A_g = r_1(I_n).$$

Due to the uniqueness of Theorem 3.5.6,

$$r_1\left(U_{(\pi)}^{-1}P\right) = (\beta_1, \ldots, \beta_n).$$

Recall that multiplication by a permutation matrix to the right has the effect of rearranging the columns, and to the left has the effect of re-arranging the rows. Since U^{-1} is also circulant, hence every row has the same weight, we obtain

$$\begin{aligned} wt(\beta_1, \ldots, \beta_n) &= wt\left(r_1(U_{(\pi)}^{-1}P)\right) = wt\left(r_1(U_{(\pi)}^{-1})\right) \\ &= wt\left(r_1(P_\pi^{-1}U^{-1})\right) = wt(r_1(U^{-1})) \\ &= wt(\alpha_1, \ldots, \alpha_n). \end{aligned}$$

□

Example 3.5.11. [56] Take $n = 5$, and $f \overset{s}{\sim} g$ whose SANFs are $x_1x_2x_4$, respectively, $x_1x_2x_3$ (and so, $wt(\Delta(f)) = wt(\Delta(g))$). Certainly,

$$A_f = C\langle 1, 1, 0, 1, 0\rangle, \qquad A_g = C\langle 1, 1, 1, 0, 0\rangle$$

$$A_f^{-1} = C\langle 0, 1, 1, 1, 0\rangle, \quad A_g^{-1} = C\langle 0, 1, 1, 0, 1\rangle,$$

and so, $wt(\Delta(f^*)) = wt(\Delta(g^*))$ (in fact, in this case the dual of f is $f^* = g$). As another example, we take $n = 8$, f, g with SANFs $x_1x_2x_4$, respectively, $x_1x_4x_5$ (and so, $wt(\Delta(f)) = wt(\Delta(g))$). We compute

$$A_f = C\langle 1, 1, 0, 1, 0, 0, 0, 0\rangle, \qquad A_g = C\langle 1, 0, 0, 1, 1, 0, 0, 0\rangle$$

$$A_f^{-1} = C\langle 1, 0, 1, 0, 0, 1, 1, 1\rangle, \quad A_g^{-1} = C\langle 0, 0, 1, 0, 0, 1, 1, 0\rangle,$$

and so, $wt(\Delta(f^*)) = 5 \neq wt(\Delta(g^*)) = 3$, therefore $f \overset{s}{\not\sim} g$.

Remark 3.5.12. The conditions $wt(\Delta(f)) = wt(\Delta(g))$, $wt(\Delta(f^*)) = wt(\Delta(g^*))$ are not sufficient to ensure that the functions f, g are S-equivalent. As an example, take $n = 8$ and f, g with $\Delta(f) = [1, 2, 3], \Delta(g) = [1, 2, 4]$. The two functions are not in the same S-equivalence class, yet $wt(\Delta(f)) = wt(\Delta(g)) = 3$ and $wt(\Delta(f^*)) = wt(\Delta(g^*)) = 5$, as one can check easily.

For a degree d MRS, whose class A_f is not invertible, let the equivalence class of the circulant pseudoinverse matrix denoted by $A_f^\dagger$ with $\Delta(A_f^\dagger) = [j_1, j_2, \ldots, j_t]$. Then the *pseudo-dual Boolean function* is

$$f^\dagger = x_{j_1}x_{j_2}\cdots x_{j_t} \oplus x_{j_1+1}x_{j_2+1}\cdots x_{j_t+1} \oplus \cdots \oplus x_{j_1-1}x_{j_2-1}\cdots x_{j_t-1}.$$

We propose the following question, which seems to be true, based on computer data.

Open Problem. [56] *If $f \overset{s}{\sim} g$ with singular matrices A_f and A_g, respectively with circulant pseudoinverses, is it true that $wt(\Delta(f)) = wt(\Delta(g))$ implies $wt(\Delta(f^\dagger)) = wt(\Delta(g^\dagger))$?*

We now present some results obtained while pursuing the open problem.

Theorem 3.5.13. [56] *Let f and g be two n-variables MRS with $f \overset{s}{\sim} g$, and $A_f = C\langle a_1, \ldots, a_n\rangle$, $A_g = C\langle a_{\pi(1)}, \ldots, a_{\pi(n)}\rangle$ for some permutation π. The matrices have pseudoinverses $C\langle \alpha_1 \ldots, \alpha_n\rangle$ and $C\langle \beta_1, \ldots, \beta_n\rangle$, respectively. Let τ be the permutation $\tau(1) = 1, \tau(2) = \lceil n/2 \rceil + 1, \tau(3) = 2, \tau(4) = \lceil n/2 \rceil + 2, \ldots$. The following statements are true:*

(i) *Let n be odd. Then*

$$
\begin{aligned}
(a_1, \ldots, a_n) &= \left(a_{\tau(1)}, \ldots, a_{\tau(n)}\right) C(\alpha_1, \ldots, \alpha_n) \\
(\alpha_1, \ldots, \alpha_n) &= \left(\alpha_{\tau(1)}, \ldots, \alpha_{\tau(n)}\right) C(a_1, \ldots, a_n) \\
\left(a_{\pi(1)}, \ldots, a_{\pi(n)}\right) &= \left(a_{(\pi\circ\tau)(1)}, \ldots, a_{(\pi\circ\tau)(n)}\right) C(\beta_1, \ldots, \beta_n) \\
(\beta_1, \ldots, \beta_n) &= \left(\beta_{\tau(1)}, \ldots, \beta_{\tau(n)}\right) C\left(a_{\pi(1)}, \ldots, a_{\pi(n)}\right).
\end{aligned}
$$

(ii) *Let n be even. Then*

$$\begin{aligned}
(a_1,\ldots,a_n) &= \left(a_{\tau(1)} \oplus a_{\tau(2)}, 0, a_{\tau(3)} \oplus a_{\tau(4)}, 0, \ldots\right) C(\alpha_1,\ldots,\alpha_n) \\
(\alpha_1,\ldots,\alpha_n) &= \left(\alpha_{\tau(1)} \oplus \alpha_{\tau(2)}, 0, \alpha_{\tau(3)} \oplus \alpha_{\tau(4)}, 0, \ldots\right) C(a_1,\ldots,a_n) \\
\left(a_{\pi(1)},\ldots,a_{\pi(n)}\right) &= \left(a_{(\pi\circ\tau)(1)} \oplus a_{(\pi\circ\tau)(2)}, 0, \ldots\right) C(\beta_1,\ldots,\beta_n) \\
(\beta_1,\ldots,\beta_n) &= \left(\beta_{\tau(1)} \oplus \beta_{\tau(2)}, 0, \ldots,\right) C\left(a_{\pi(1)},\ldots,a_{\pi(n)}\right).
\end{aligned}$$

Proof. (i) Let n be odd. For the first part, by the definition of pseudoinverse,

$$\begin{aligned}
(a_1,\ldots,a_n) &= (1,0,\ldots,0)C(a_1,\ldots,a_n) \\
&= (1,0,\ldots,0)C(a_1,\ldots,a_n)C(\alpha_1,\ldots,\alpha_n)C(a_1,\ldots,a_n) \\
&= (1,0,\ldots,0)C(a_1,\ldots,a_n)^2C(\alpha_1,\ldots,\alpha_n).
\end{aligned}$$

Let P_τ be the permutation matrix for τ. By Corollary 3.4.5,

$$\begin{aligned}
C(a_1,\ldots,a_n)^2 &= C(a_1, a_{\lceil n/2\rceil+1}, a_2, a_{\lceil n/2\rceil+2}, \ldots, a_{\lceil n/2\rceil}) \\
&= C((a_1,\ldots,a_n)P_\tau) \\
&= C(a_{\tau(1)},\ldots,a_{\tau(n)}).
\end{aligned}$$

Therefore,

$$\begin{aligned}(a_1,\ldots,a_n) &= ((1,0,\ldots,0)C(a_1,\ldots,a_n)^2C(\alpha_1,\ldots,\alpha_n)\\ &= (1,0,\ldots,0)C(a_{\tau(1)},\ldots,a_{\tau(n)})C(\alpha_1,\ldots,\alpha_n)\\ &= (a_{\tau(1)},\ldots,a_{\tau(n)})C(\alpha_1,\ldots,\alpha_n).\end{aligned}$$

The second part is immediate since $C(\alpha_1,\ldots,\alpha_n)$ is a pseudoinverse of $C(a_1,\ldots,a_n)$, which shows that it is also reflexive inverse.

For the third part, let P_π be the permutation matrix for π. Then, using Corollary 3.4.5,

$$\begin{aligned}\left(a_{\pi(1)},\ldots,a_{\pi(n)}\right) &= (1,0,\ldots,0)C\left(a_{\pi(1)},\ldots,a_{\pi(n)}\right)\\ &= (1,0,\ldots,0)C\left(a_{\pi(1)},\ldots,a_{\pi(n)}\right)^2C(\beta_1,\ldots,\beta_n)\\ &\left(a_{(\pi\circ\tau)(1)},\ldots,a_{(\pi\circ\tau)(n)}\right)C(\beta_1,\ldots,\beta_n).\end{aligned}$$

The fourth part is immediate, since $C(\beta_1,\ldots,\beta_n)$ is a reflexive inverse of $C(a_{\pi(1)},\ldots,a_{\pi(n)})$.

(ii) We can show this using similar techniques used in (i) with Corollary 3.4.5. □

For an MRS function f, when A_f does not have a pseudoinverse, but circulant generalized inverses, the notion of dual is not well defined. Often, the weights of the generalized inverses differ and the generalized inverses are not unique. However, they

do correspond to a unique generalized inverse, which is the smallest in lexicographical order, via the congruence modulo the corresponding H's in Theorem 3.5.6. This uniqueness is not readily recognizable in matrix form. Let us define the dual Boolean function corresponding to that unique representative of all generalized inverses. Using this notion, for singular A_f and A_g without a pseudoinverse, but with circulant generalized inverses, the condition $wt(\Delta(f^*)) = wt(\Delta(g^*))$ does not hold. To illustrate this, let $n = 7$. We check that $f = x_1x_2x_3x_5(SANF)$ and $g = x_1x_2x_3x_6(SANF)$ are S-equivalent. The functions do not have pseudoinverses, but circulant general inverses. We computed all generalized inverses that are circulant, and they are in the classes $A_f^* = C\langle 1,0,0,0,0,0,0\rangle$ and $A_g^* = C\langle 1,1,0,0,0,0,0\rangle$, respectively. Clearly, we have

$$wt(\Delta(f^*)) \neq wt(\Delta(g^*)).$$

We now consider the case of a converse of our previous theorem. For simplicity, we assume all indices are $\mod n$. Let P and Q be permutation matrices. Then, it is known that if two circulant matrices A and B are P-Q equivalent, that is, $PA = BQ$, then AA^T and BB^T are *similar matrices* [63]. Moreover, it is straightforward to see that $AA^T = \sum_{i,j\in\Delta(A)} G^{a_i-a_j}$, where $A = C(a_1,\ldots,a_n)$. This actually points to the importance of the differences $a_i - a_j$, which played a role in Cusick's paper [55], which only addresses the MRS functions with $wt(\Delta(f)) = 3$. Given a permutation δ, we let P_δ be the row permutation matrix corresponding to the permutation δ.

Theorem 3.5.14. [56] *Let f and g be MRS functions with $A_f = C(a_1,\ldots,a_n)$, $A_g = C(b_1,\ldots,b_n)$, respectively. Let a permutation matrices P_σ for the permutation σ and a permutation matrix Q_τ for the permutation τ such that $P_\sigma A_f = A_g Q_\tau$. Then, $wt(\Delta(f)) = wt(\Delta(g))$ and $a_{\sigma(j)+i-1} = b_{\tau(i)+j-1}$.*

If A_f and B_g are invertible, then we also have

$$(\alpha_1,\ldots,\alpha_n) = \left(\beta_{\sigma^{-1}(1)-\tau(1)+n+1},\ldots,\beta_{\sigma^{-1}(n)-\tau(1)+n+1}\right),$$

and

$$wt(\Delta(f^*)) = wt(\Delta(g^*))$$

where $(\alpha_1, \ldots, \alpha_n) = A_f^{-1}$ *and* $(\beta_1, \ldots, \beta_n) = A_g^{-1}$.

Proof. Let $A_f = C(a_1, \ldots, a_n)$ and $A_g = C(b_1, \ldots, b_n)$. We write

$$\begin{aligned} P_\sigma A_f &= \begin{pmatrix} a_{\sigma(1)} & a_{\sigma(1)+1} & \cdots & a_{\sigma(1)+n-1} \\ a_{\sigma(2)} & a_{\sigma(2)+1} & \cdots & a_{\sigma(2)+n-1} \\ \cdot & \cdots & \cdots & \cdots \\ a_{\sigma(n)} & a_{\sigma(n)+1} & \cdots & a_{\sigma(n)+n-1} \end{pmatrix} \\ A_g Q_\tau &= \begin{pmatrix} b_{\tau(1)} & b_{\tau(2)} & \cdots & b_{\tau(n)} \\ b_{\tau(1)+1} & b_{\tau(2)+1} & \cdots & b_{\tau(n)+1} \\ \cdots & \cdots & \cdots & \cdots \\ b_{\tau(1)+n-1} & b_{\tau(2)+n-1} & \cdots & b_{\tau(n)+n-1} \end{pmatrix}. \end{aligned}$$

From $P_\sigma A_f = A_g Q_\tau$, we derive

$$a_{\sigma(j)+i-1} = b_{\tau(i)+j-1}.$$

We note that the first rows of $P_\sigma A_f$ and $A_g Q_\tau$ are the same. Also, the sets $\{\sigma(1), \sigma(1)+1, \ldots, \sigma(1)+n-1\}$ and $\{\tau(1), \tau(2), \ldots, \tau(n)\}$ are simply permutations of $\{1, 2, \ldots, n\}$. Therefore, we see that

$$wt\left(a_{\sigma(1)}, a_{\sigma(1)+1}, \ldots, a_{\sigma(1)+n-1}\right) = wt(a_1, a_2, \ldots, a_n),$$

$$wt\left(b_{\tau(1)}, b_{\tau(2)}, \ldots, b_{\tau(n)}\right) = wt(b_1, b_2, \ldots, b_n),$$

and

$$wt(\Delta(f)) = wt(\Delta(g)).$$

From Theorem 3.5.6, α_i and β_i with $1 \leq i \leq n$ are unique with the property

$$\begin{aligned} (1,0,\ldots,0) &= (\alpha_1,\ldots,\alpha_n)C(a_1,\ldots,a_n) \\ (1,0,\ldots,0) &= (\beta_1,\ldots,\beta_n)C(b_1,\ldots,b_n). \end{aligned} \tag{3.5}$$

We multiply the second relation by Q_τ from the right and obtain

$$\begin{aligned} (0,\ldots,0,\overset{\tau(1)}{\underset{\downarrow}{1}},0,\ldots) &= (\beta_1,\ldots,\beta_n)A_gQ_\tau \\ &= (\beta_1,\ldots,\beta_n)P_\sigma A_f \\ &= \left(\beta_{\sigma^{-1}(1)},\ldots,\beta_{\sigma^{-1}(n)}\right)A_f. \end{aligned} \tag{3.6}$$

We multiply the last equation from the right by the permutation matrix $R_{\rho^{n+1-\tau(1)}}$, corresponding to the shift $\rho^{n+1-\tau(1)}$, to rewrite the left hand side of (3.6) in the standard form $(1,0,\ldots,0)$. Since $R_{\rho^{n+1-\tau(1)}}$ is also a circulant matrix, by Lemma 3.4.4, it will commute with A_f and (3.6) becomes

$$\begin{aligned} (1,0,\ldots,0) &= \left(\beta_{\sigma^{-1}(1)},\ldots,\beta_{\sigma^{-n}(1)}\right)R_{\rho^{n+1-\tau(1)}}A_f \\ &= \left(\beta_{\sigma^{-1}(1)-\tau(1)+1},\ldots,\beta_{\sigma^{-1}(n)-\tau(1)+1}\right)A_f. \end{aligned}$$

Since $(\alpha_1,\ldots,\alpha_n)$ was unique with the property (3.5),

$$(\alpha_1,\ldots,\alpha_n) = \left(\beta_{\sigma^{-1}(1)-\tau(1)+1},\ldots,\beta_{\sigma^{-1}(n)-\tau(1)+1}\right)$$

where the indices are $\mod n$. Since the indices above right are just a permutation of $\{1, 2, \ldots, n\}$, we immediately get $wt(\Delta(f^*)) = wt(\Delta(g^*))$.

□

The previous theorem easily extends to the following corollary.

Corollary 3.5.15. [56] *Let f and g be two full–cycle MRS functions with the invertible classes A_f and A_g, respectively. Let $A_f^{-1} = C\langle \alpha_1, \ldots, \alpha_n \rangle$ and $A_g^{-1} = C\langle \beta_1, \ldots, \beta_n \rangle$. If $f \overset{s}{\sim} g$, then there exists a permutation matrix P such that*

$$P \cdot (\alpha_1, \ldots, \alpha_n)^T = (\beta_1, \ldots, \beta_n)^T.$$

THIS PAGE INTENTIONALLY LEFT BLANK

4. MRS BOOLEAN FUNCTIONS AND GRAPHS

4.1. INTRODUCTION

The difficulty in the affine equivalence problem may be mitigated by establishing relationships to other disciplines in mathematics for possible solutions. Graph theory studies the properties of a graph, which is a structure defined by a set of vertices (or nodes) and a set of edges which connect vertices to each other. Often, a graph representation of an algebraic structure helps us to visualize the complexity of the structure. One simple example is visualization of a Boolean function using a tree, which is a graph in which each pair of vertices is connected by a unique path. There have been many attempts to establish meaningful relationships between graphs and Boolean functions. One of the interesting connections involves bent functions and Cayley graphs. In [64], Bernasconi and Condenotti showed that the Walsh transforms of some Boolean functions can be analyzed by a Cayley graph representation of Boolean functions. They later extended their finding to the characterization of bent functions, using strongly regular graphs in [65]. In 2007, Stanica [66] presented necessary conditions for bent functions and investigated the propagation criteria of Boolean functions, using the Cayley graph representation. In this chapter, we present some basic graph-theory material, briefly review the Cayley graph representation, and present a new graph representation of MRS functions and some analysis in regard to S-equivalence.

4.2. EXAMPLE OF GRAPH REPRESENTATION OF BOOLEAN FUNCTIONS

4.2.1. Definitions and Fundamentals of a Graph

A *graph* $G = (V,\, E)$ is defined by a set of *vertices*, V or $V(G)$ and a set of *edges,* E or $E(G) = \{\{x, y\} \,|\, x,\, y \in V, \text{and}\, x \neq y\}$. If $\{x, y\} \in E(G)$, we say that x and y are adjacent. The number of edges that are incident with the vertex v is the *degree* of v, denoted by $deg(v)$. Two vertices are connected if we can go from one vertex to the

other by traveling a path defined by the edges of the graph. A graph is *connected* if for every pair of vertices, there exists a path of edges connecting them. If a graph is not connected, it is disconnected. If each vertex of a graph G has the same degree, we call G a *regular* graph. A regular graph G is *strongly regular* if there exist two integers m and n such that every two adjacent vertices have m common neighbors, and every two nonadjacent vertices have n common neighbors. A graph G is *bipartite* if $V(G)$ can be partitioned into two sets V_1 and V_2 such that there exists no edge $\{v, w\}$ with $v,\ w \in V_1$ or $v,\ w \in V_2$. A graph G is *complete* if $E(G)$ contains all possible edges. We denote the complete graph on n vertices by K_n. Another special graph we use in this chapter is a *cycle*. In this thesis, we denote a cycle as $[v_1, v_2, \ldots, v_n]$ where $\{v_1, v_2, \ldots v_n\} \subseteq V(G)$ and $E = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots \{v_{n-1}, v_n\}, \{v_n, v_1\}\}$. Clearly, a cycle is a connected regular graph (or subgraph) of degree 2. Next, we give a formal definition of equality and isomorphism in graphs.

Definition 4.2.1. Two graphs $G(V_G, E_g)$ and $H(V_H, E_H)$ are *equal* if

$$V_G = V_H \text{ and } E_G = E_H.$$

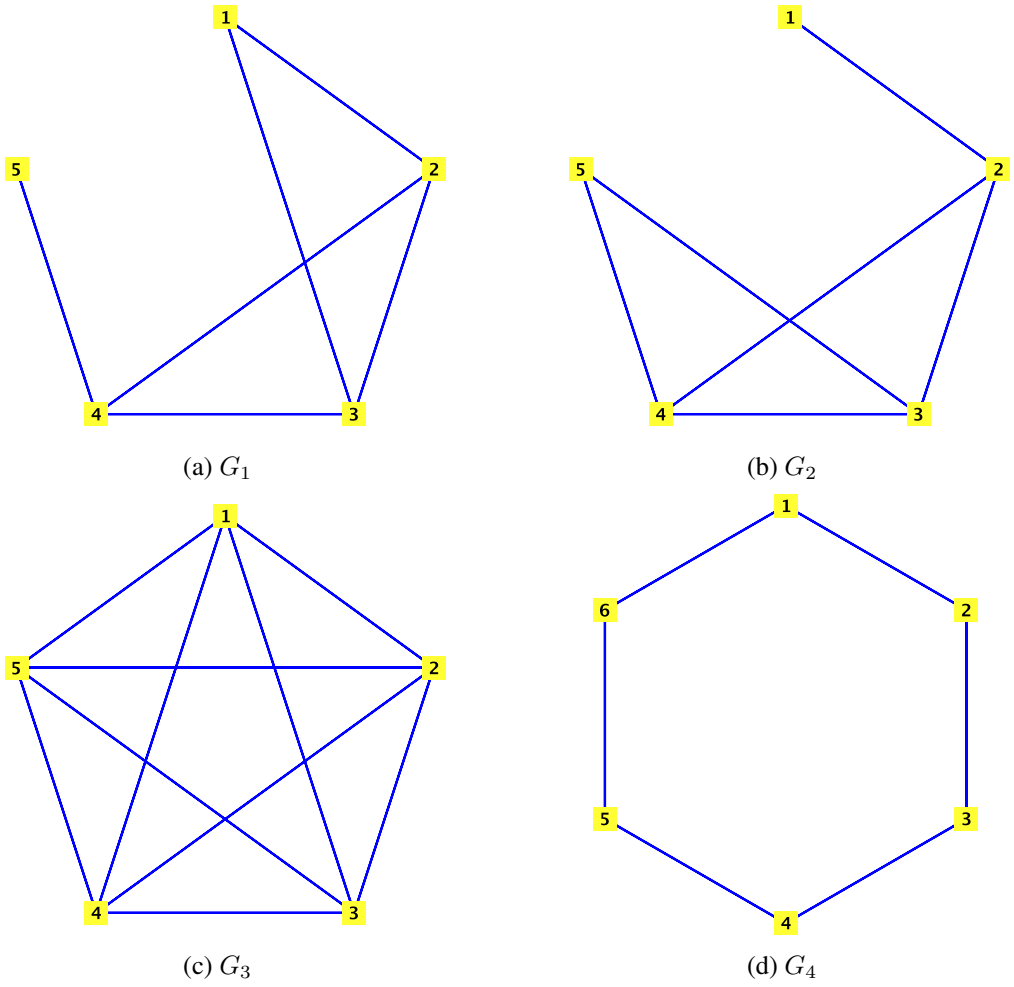
The graphs G and H are *isomorphic* if there exists a bijection

$$f : V_G \to V_H,$$

such that for any vertices $u, v \in V_G$, $\{u, v\} \in E_G$ if and only if $\{f(u), f(v)\} \in E_H$.

Example 4.2.2. Let G_1 be the graph with $V(G_1) = \{1, 2, 3, 4, 5\}$ and $E(G_1) = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{4, 5\}\}$. Sub-figure (a) of Figure 4.1 represents a drawing of G_1. The graph G_1 is not regular, since $deg(1) = 2$ and $deg(2) = 3$. The graphs G_1 and G_2 are isomorphic by the permutation $(1, 5)(2, 4)$. The graph G_3 is K_5 and clearly strongly regular. The graph G_4 on the sub-figure (d) is the cycle $[1, 2, 3, 4, 5, 6]$. However, it is not strongly regular, since the vertices 1 and 3 have one common neigh-

bor 2, but vertices 1 and 4 have no common neighbor. It is bipartite, with the partition $V_1 = \{1, 3, 5\}$ and $V_2 = \{2, 4, 6\}$.



(a) G_1 (b) G_2

(c) G_3 (d) G_4

Figure 4.1: Simple Graphs

4.2.2. An Example of Application of Graph Theory to Cryptographic Boolean Function

There have been many attempts to establish relationships between graph theory and Boolean functions. One of the most interesting relationships involves affine equivalence of Boolean functions and Cayley graphs.

Definition 4.2.3. [4, p. 194] Let f be a Boolean function of n variables. The Cayley graph of f, denoted by $\Gamma_f = (V, E)$, is defined by $V = \mathbb{F}_2^n$ and

$$E = \{\{v, w\} \mid v,\, w \in \mathbb{F}_2^n,\, v \neq w, \text{ and } f(v \oplus w) = 1\}.$$

In [64], Bernasconi and Codenotti introduced the relationship between the Cayley graph representation of Boolean functions and affine equivalent classes of four variable Boolean functions. They established an isomorphism between the eight affine equivalent classes of the 4-variable Boolean functions and eight classes of regular graphs with 16 vertices. Table 4.1 and Figure 4.2 illustrate their findings. They observed that, as the nonlinearity increases in the affine equivalent classes, the degree and connectivity of the matching graphs increase as well. Notably, Class V and VI graphs are degree 4-regular graphs, but Class VI graph is connected, whereas Class V is disconnected. A supplemental analysis of the relationship and other related materials can be found in [4, pp. 205–208].

4.3. A GRAPH REPRESENTATION OF ROTATION-SYMMETRIC BOOLEAN FUNCTIONS

We recall that an MRS function has a cyclical structure in its algebraic normal form (ANF). Adopting this feature, we attempt to represent a Boolean function with a graph with a similar property. We observe that an MRS function is a homogeneous function where each multiplication term of variables can be represented as a cycle. For example, the MRS Boolean function $f(\mathbf{x}) = x_1x_2x_3 \oplus x_2x_3x_4 \oplus x_3x_4x_5 \oplus x_4x_5x_6 \oplus x_5x_6x_1 \oplus x_6x_1x_2$ of six variables can generate six cycles on vertices 1 through 6, that is $[1, 2, 3]$, $[2, 3, 4]$, $[3, 4, 5]$, $[4, 5, 6]$,$[5, 6, 1]$, and $[6, 1, 2]$. We can combine them, disregarding multiple edges, and obtain the graph represented in Figure 4.3. We note that the graph is regular but not strongly regular, since non-neighboring vertices 1 and 3 have the common neighbors vertices 2 and 5, but 1 and 4 have the common neighbors 2, 3, 5, and 6.

However, this construction may present a problem with the ordering of variables. Consider the following example.

Class	Boolean Function
I	0000000000000000
II	0000000000000001
III	0000000000000011
IV	0000000000000111
V	0000000000001111
VI	0000000000010111
VII	0000000100010111
VIII	0000001101011001

Class	Walsh Spectrum															
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
II	1	-1	-1	1	-1	1	1	-1	-1	1	1	-1	1	-1	-1	1
III	2	0	-2	0	-2	0	2	0	-2	0	2	0	2	0	-2	0
IV	3	-1	-1	-1	-3	1	1	1	-3	1	1	1	3	-1	-1	-1
V	4	0	0	0	-4	0	0	0	-4	0	0	0	4	0	0	0
VI	4	-2	-2	0	-2	0	0	2	-4	2	2	0	2	0	0	-2
VII	5	-3	-3	1	-3	1	1	1	-3	1	1	1	1	1	1	-3
VIII	6	-2	-2	2	-2	-2	2	-2	-2	2	-2	-2	-2	2	2	2

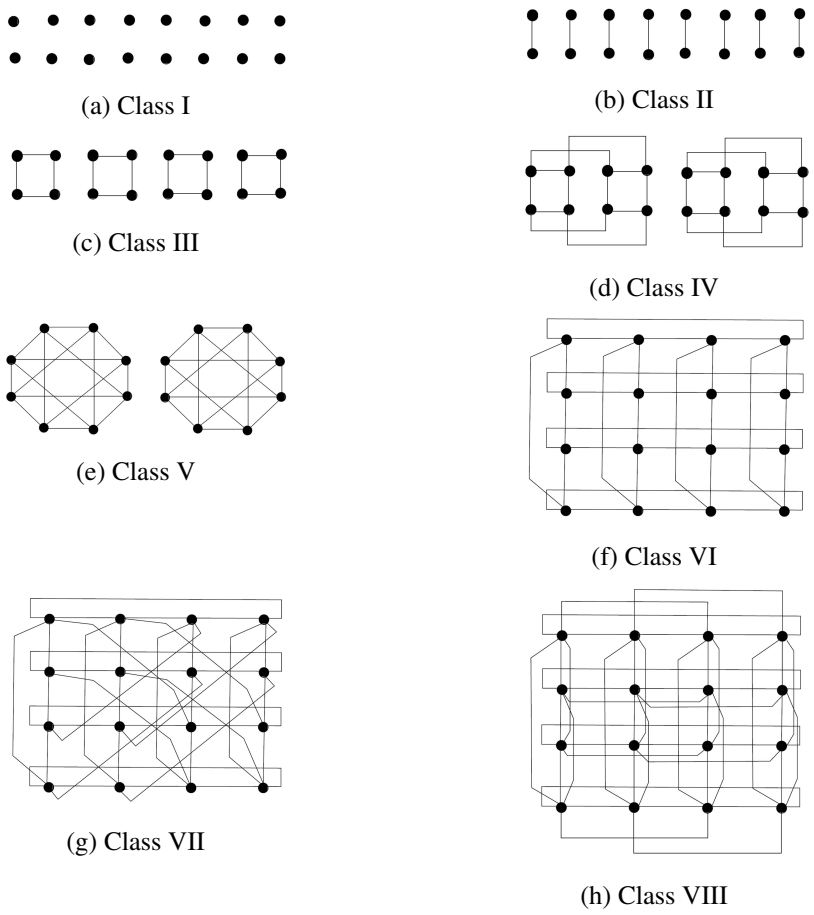
Table 4.1: Affine Equivalence Classes of 4-Variable Boolean Functions From [64]

Example 4.3.1. Let $f = x_1x_2x_3x_4(SANF) \in \mathbb{F}_2^6$. Algebraically, $x_1x_2x_3x_4 = x_1x_3x_2x_4$. However, they generate two different cycles and hence two different graph representations as shown in Figure 4.4.

This indicates that the cyclic representation of MRS is sensitive to the order of variables. In order to obtain a consistent graph not affected by this ordering problem, we introduce the following notion, adding an order property to the definition of SANF.

Definition 4.3.2. Let f be an MRS function of n variables with the SANF $x_{j_1}x_{j_2}\cdots x_{j_d}$, where $1 \leq d \leq n$. The *ordered short algebraic normal form (OSANF)* of f, denoted by $f = x_1x_{i_2}\cdots x_{i_d}(OSANF)$ or $f = \|x_1x_{i_2}\cdots x_{i_d}\|$ is the SANF $x_{i_1}x_{i_2}\cdots x_{i_d}$ such that $i_1 = 1$ and $1 = i_1 < i_2 < \cdots < i_d$.

By Definition 4.3.2, our scheme generates one and only one graph for each MRS Boolean function.



(a) Class I

(b) Class II

(c) Class III

(d) Class IV

(e) Class V

(f) Class VI

(g) Class VII

(h) Class VIII

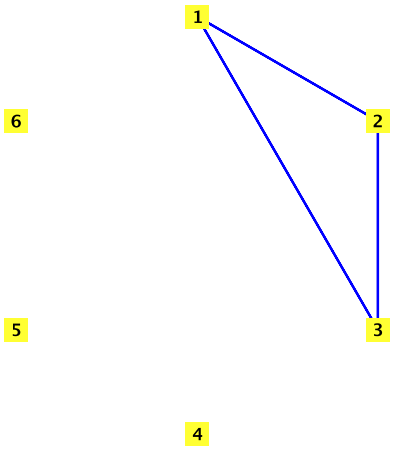
Figure 4.2: Cayley Graph Classes of 4-Variable Boolean Function From [64]

Definition 4.3.3. A *cycle combination graph* (CCG) of an n-variable MRS Boolean function $f(\mathbf{x}) = x_1 x_{P_2} x_{P_3} \dots x_{P_d} (OSANF)$ with $d \leq n$, denoted by G_f is a simple graph with $V = \{1, 2, \dots n\}$ and the edges of the cycles,

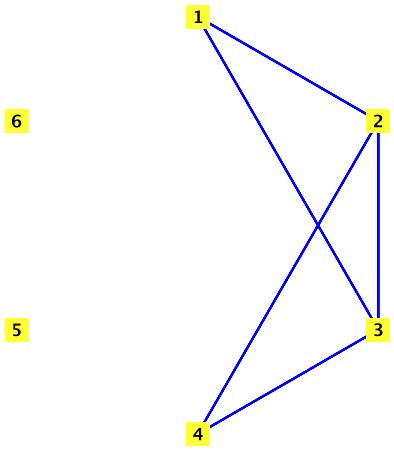
$$[1, P_2, P_3, \dots, P_d]$$

$$[2, P_2 + 1 \bmod n, P_3 + 1 \bmod n, \dots, P_d + 1 \bmod n], \text{ and}$$

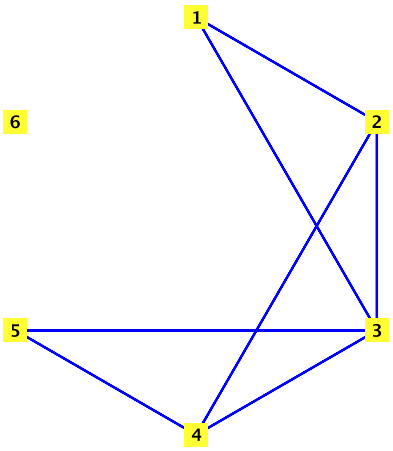
$$[n, P_2 + n - 1 \bmod n, P_3 + n - 1 \bmod n, \dots, P_d + n - 1 \bmod n,],$$



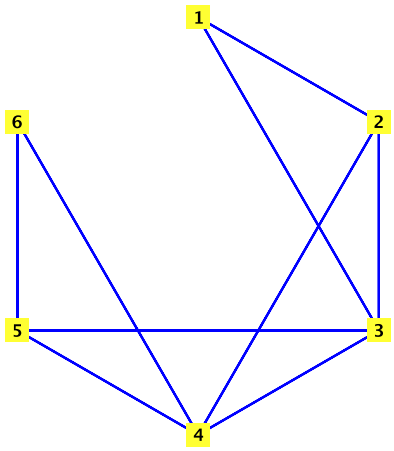
(a) Cycle 1



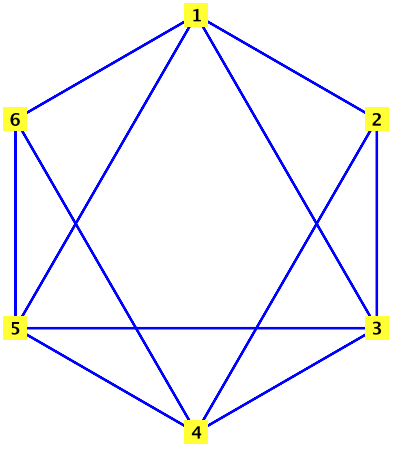
(b) Cycles 1 and 2



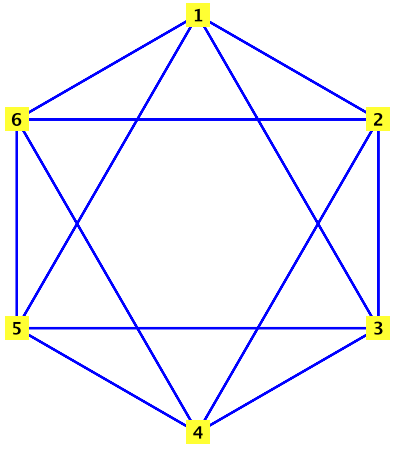
(c) Cycles 1 - 3



(d) Cycles 1 - 4

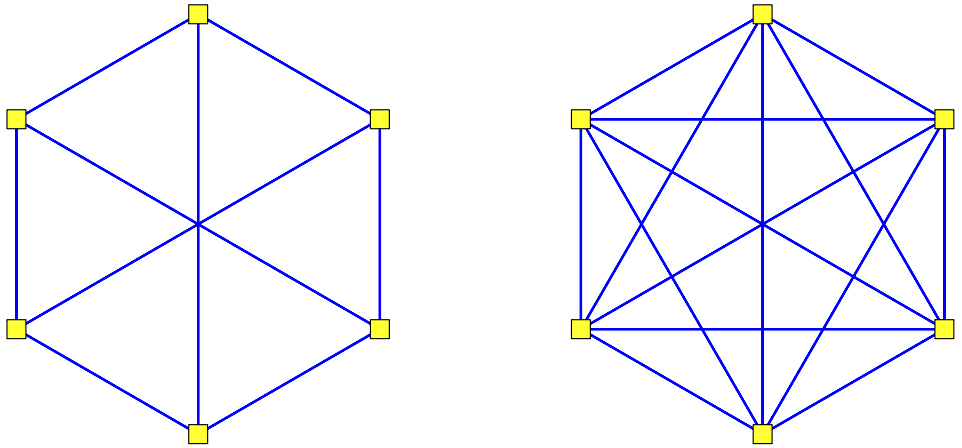


(e) Cycles 1 - 5



(f) Cycles 1 - 6

Figure 4.3: A Cycle Combination of an MRS Boolean Function



(a) $x_1x_2x_3x_4(SANF)$ (b) $x_1x_3x_2x_4(SANF)$

Figure 4.4: Two Graphs Generated by the Same SANF

regarding multiple edges as one edge.

Remark 4.3.4. In order to make our algebraic operations for the indices work, we add an additional property to the modular arithmetic in this chapter.

We set

$$n \bmod n = 0.$$

This gives us $x_0 = x_n$, and we use the notations interchangeably.

We observe that two Boolean functions in $\mathcal{B}_n$ form a relationship with respect to the CCG. The relationship satisfies reflexivity, symmetry, and transitivity. Therefore, it is an equivalence relation and partitions the Boolean functions of n variable into equivalence classes.

Definition 4.3.5. Two MRS functions of same variables f and h are *cycle combination graph (CCG) equivalent*, denoted by $f \overset{C}{\sim} h$ if G_f is isomorphic to G_h.

MRS functions add interesting characteristics to the structure of CCGs. These characteristics originate from the cycles generated by shifting variables. Table 4.2 illustrates how shifting along the indices of the variables effect the cycles.

Rotation	Vertex Index Shift			
$P_1 = 1$	P_2	P_3	$\cdots\cdots$	P_d
2	$P_2 + 1 \bmod n$	$P_3 + 1 \bmod n$	$\cdots\cdots$	$P_d + 1 \bmod n$
3	$P_2 + 2 \bmod n$	$P_3 + 2 \bmod n$	$\cdots\cdots$	$P_d + 2 \bmod n$
$\vdots$	$\vdots$	$\vdots$		$\vdots$
m	$P_2 + m - 1 \bmod n$	$P_3 + m - 1 \bmod n$		$P_d + m - 1 \bmod n$
$\vdots$	$\vdots$	$\vdots$	$\cdots\cdots$	$\vdots$
$n - 1$	$P_2 + n - 2 \bmod n$	$P_3 + n - 2 \bmod n$	$\cdots\cdots$	$P_d + n - 2 \bmod n$
n	$P_2 + n - 1 \bmod n$	$P_3 + n - 1 \bmod n$	$\cdots\cdots$	$P_d + n - 1 \bmod n$

Table 4.2: Vertex Structure of a Cycle Combination Graph of a MRS Function

In order to analyze what happens at each vertex, we measure the distance from each variable in the monomial term to x_n. Let k_i be the distance from x_{P_i} to x_n defined by $k_i = n - P_i$. Therefore, we have

$$\begin{aligned} k_1 &= n - 1, \\ k_2 &= n - P_2, \\ &\vdots \\ &\vdots \\ k_d &= n - P_d. \end{aligned}$$

Additionally, since we are working with the cycles derived from the variables of a Boolean function in ANF, we can measure the distance between the vertices in the following manner. Let r_i be the distance between $x_{P_{i+1}}$ and x_{P_i} defined by $r_i = P_{i+1} - P_i$. Then, we have

$$
\begin{aligned}
r_1 &= P_2 - P_1, \\
r_2 &= P_3 - P_2, \\
&\vdots \\
&\vdots \\
r_d &= P_{d+1} - P_d,
\end{aligned}
\tag{4.1}
$$

where $P_{d+1} = n + 1$.

We focus on vertex 1. Vertex 1 connects $2d$ times, as shown in Table 4.3.

Shift	Vertex 1 and its Neighbors by Shift		
0	P_d	$P_1 = 1$	P_2
1	$P_1 + k_2 + 1 \bmod n$	$P_2 + k_2 + 1 \bmod n = 1$	$P_3 + k_2 + 1 \bmod n$
2	$P_2 + k_3 + 1 \bmod n$	$P_3 + k_3 + 1 \bmod n = 1$	$P_4 + k_3 + 1 \bmod n$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
d-1	$P_{d-1} + k_d + 1 \bmod n$	$P_d + k_d + 1 \bmod n = 1$	$P_1 + k_d + 1 \bmod n$

Table 4.3: Vertex 1 and its Neighbors

By applying the descriptions of k_i and r_i with $1 \leq i \leq d$, we see that a set of edges on vertex $P_1 = 1$, as justified below:

$$
\begin{aligned}
&\{1,\ 1 \pm r_1 \pmod n\} \\
&\{1,\ 1 \pm r_2 \pmod n\} \\
&\qquad\vdots \\
&\qquad\vdots \\
&\{1,\ 1 \pm r_d \pmod n\}.
\end{aligned}
\tag{4.2}
$$

By the shifting action of the CCG generation, the set of edges replicates on each vertex, depending only on r_i's. Therefore, by an inductive argument, we can generalize the result for any vertex. Table 4.4 shows the neighbors of an arbitrary vertex m.

Shift	Neighbors of Vertex m		
0	$P_d + m - 1 \bmod n$	m	$P_2 + m - 1 \bmod n$
1	$P_1 + k_2 + m \bmod n$	$P_2 + k_2 + m \bmod n = m$	$P_3 + k_2 + m \bmod n$
2	$P_2 + k_3 + m \bmod n$	$P_3 + k_3 + m \bmod n = m$	$P_4 + k_3 + m \bmod n$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
d-1	$P_{d-1} + k_d + m \bmod n$	$P_d + k_d + m \bmod n = m$	$P_1 + k_d + m \bmod n$

Table 4.4: $2d$ Neighbors of Arbitrary Vertex m

Applying the same argument as for the vertex 1, we obtain the following neighbors

$$\begin{aligned}
&\{m,\ m \pm r_1 \pmod n\},\\
&\{m,\ m \pm r_2 \pmod n\},\\
&\qquad\vdots\\
&\qquad\vdots\\
&\{m,\ m \pm r_d \pmod n\}.
\end{aligned} \tag{4.3}$$

This generalization suggests that the CCGs are regular, since a CCG is a simple graph.

Theorem 4.3.6. *Let f be an MRS function of n variables generated by $x_1 x_{p_2} \dots x_{p_d}(OSANF)$ and G_f be the CCG of f. Then G_f is regular.*

In particular, G_f is

$$\left|\{\{1, 1 \pm r_i \pmod n\} \mid 1 \le i \le d\}\right| \textit{-regular},$$

where r_i are defined as in Equation 4.1.

Proof. By Equation 4.3, vertex 1 has $2d$ many edges, counting multiple edges, and the cardinality of

$$\{\{1, 1 \pm r_i \pmod n\} \mid 1 \leq i \leq d\}$$

gives us the number of edges at each vertex, counting multiple edges as one. Also, the degree of a vertex does not depend on the vertex, as discussed. Therefore, the claim holds. □

Generally, each distinct r_i adds two edges to a vertex, except when the two edges coincide with each other. We see that the exception results in an r-regular graph, where r is an odd number.

Corollary 4.3.7. *Let $f = x_1 x_{p_2} \dots x_{p_d}(OSANF)$ be a MRS function of n variables. Then, G_f is t-regular graph where $t = 2k_1 - 1$ for some $k_1 \in \mathbb{N}$ if and only if $n = 2k_2$ for some integer $k_2 \in \mathbb{N}$, and there exists i with $1 \leq i \leq$d such that $r_i = k_2$.*

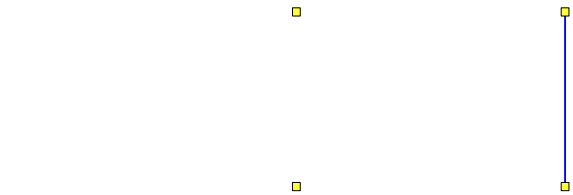
Proof. ($\Leftarrow$) In line with Theorem 4.3.6, for an arbitrary vertex m, we have two edges $\{m, m + k_2 \bmod n\}$ and $\{m, m - k_2 \bmod n\}$. Since $n = 2k_2$,

$$m + k_2 \bmod n = m - k_2 \bmod n.$$

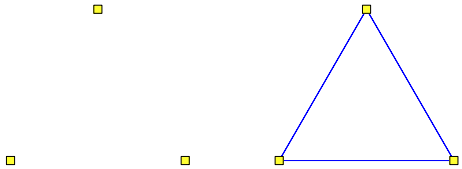
Hence, $r_i = k_2$ adds one edge to G_f. Additionally, any $r_i \neq k_2$ adds two edges to G_f. Therefore, G_f is t-regular graph where $t = 2k_1 - 1$ for some $k_1 \in \mathbb{N}$.

($\Rightarrow$) First, we claim n is even. If n is odd, Theorem 4.3.6 implies that each r_i adds two distinct edges to a vertex. This contradicts that t is odd. In addition, if $r_i \neq k_2$ for all i, then we see that r_i's add two edges to the vertex, which makes t even, a contradiction. Therefore, the claim holds. □

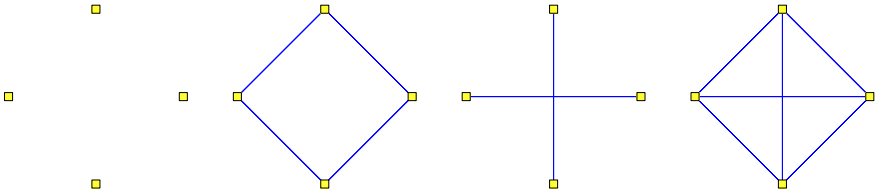
Using Table 4.2, we generate some possible configurations of graphs for MRS functions in Figure 4.5. They suggest that the CCGs for the functions of the order greater than three are generated by the union of CCGs of quadratic functions. However, when $n = 5$, the



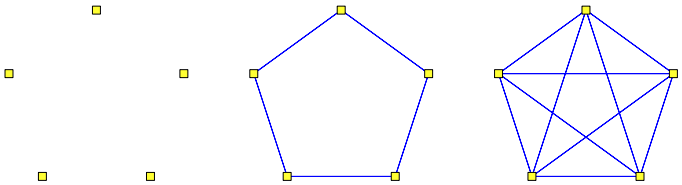
(a) $n = 2$



(b) $n = 3$



(c) $n = 4$



(d) $n = 5$

Figure 4.5: Isomorphic Cycle Combination Graph Classes $n = 2$ to 5

CCG K_5 is generated by two cycles $[1, 2, 3, 4, 5]$ and $[1, 3, 5, 2, 4]$, which are the CCGs of $x_1x_2(OSANF)$ and $x_1x_3(OSANF)$, respectively, and they are isomorphic to each other. This shows that generating quadratic functions may be isomorphic in their CCGs. Furthermore, Equation 4.3 suggests that we get a pair of edges from a quadratic function, which generates the CCG by shifting n times through the vertices. This implies that the space of

CCGs for n variable Boolean functions can be generated by the CCGs of quadratic functions,

$$x_1x_2(OSANF), x_1x_3(OSANF), \ldots, \text{ and } x_1x_{\lfloor \frac{n}{2} \rfloor}(OSANF).$$

Therefore, given n variable MRS Boolean functions, the maximum number of possible CCGs is

$$\sum_{i=0}^{\lfloor \frac{n}{2} \rfloor} \binom{\lfloor \frac{n}{2} \rfloor}{i} = 2^{\lfloor \frac{n}{2} \rfloor}.$$

This gives us the following lemma.

Lemma 4.3.8. *Given $n \in \mathbb{N}$, the maximum number of CCGs of an n-variable MRS is bounded above by $2^{\lfloor \frac{n}{2} \rfloor}$.*

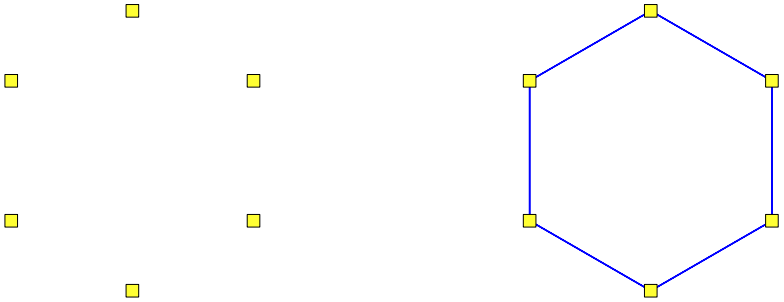
The bound in Lemma 4.3.8 cannot improve to equality, since we have cases where some unions of the quadratic CCGs are impossible under certain conditions. We illustrate this in the following example.

Example 4.3.9. In Figure 4.6, the sub-figures (b) through (d) form a basis for the graph space for $n = 6$, which generates the rest of the CCG's, the sub-figures (e) through (g).

We note that the configuration in Figure 4.7 is not a possible CCG. The graph is a combination of G_c and G_d in Figure 4.6. Therefore, we have to use the edges connecting two numbers apart by 2 or 3. This implies that we cannot complete a cycle in Figure 4.7 without violating the order structure of CCG. In other words, it is equivalent to a partition on six identical objects with parts of two and three only, which is impossible. So far, we focused on the fact that the difference between the indices of variables generate two edges at a vertex of the CCG. We note that we just need one of the two edges, and so we can simplify the notion with the next definition.

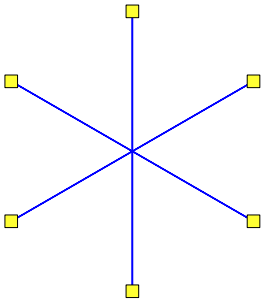
Definition 4.3.10. Let $f = x_1x_{P_2}x_{P_3}\ldots x_{P_d}(OSANF)$. Let r_i be as in Equation 4.1.

The *distance set* of f, denoted by $DS(f)$, is the set

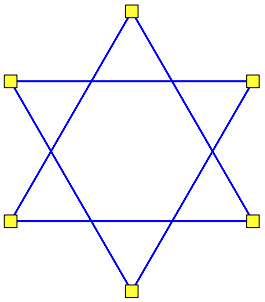


(a) G_a

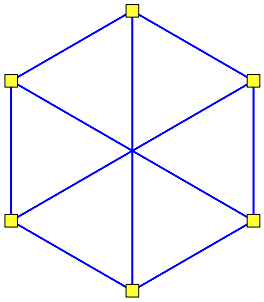
(b) G_b



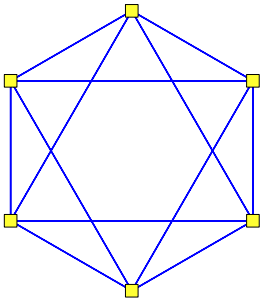
(c) G_c



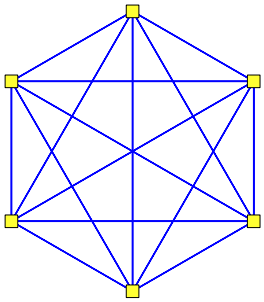
(d) G_d



(e) G_e



(f) G_f



(g) G_g

Figure 4.6: Cycle Combination Graphs $n = 6$

$$\{a_i | a_i = \min(r_i, n - r_i),\ 1 \leq i \leq d\}\,.$$

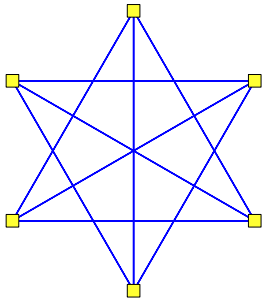


Figure 4.7: An Impossible CCG $n = 6$

We call a_i a *distance element* of f.

It is clear that each r_i generates, at most, one distance element.

Lemma 4.3.11. *Let f be an MRS function of n variables whose CCG is an r-regular graph. Then,*

$$|DS(f)| = \left\lceil \frac{r}{2} \right\rceil .$$

Proof. If n is odd, by the construction of CCG and Definition 4.3.10, each distance element generates two edges for a vertex of G_f, and so

$$|DS(f)| = \frac{r}{2} = \left\lceil \frac{r}{2} \right\rceil .$$

However, if n is even, we consider two cases. If r is even, by the construction of CCG and Definition 4.3.10, each distance element generates two edges for a vertex of G_f, and so

$$|DS(f)| = \frac{r}{2} = \left\lceil \frac{r}{2} \right\rceil .$$

If r is odd, by Corollary 4.3.7, we know

$$\frac{n}{2} \in DS(f),$$

and the distance element $\frac{n}{2}$ generates only one edge (or two overlapping edges) while each of the other distance elements generates two edges. So, we have,

$$|DS(f)| = \left\lceil \frac{r}{2} \right\rceil .$$

□

One of the characteristics of a quadratic MRS function f is that $|DS(f)| = 1$. However, not every MRS function f with $|DS(f)| = 1$ is a quadratic function. The next lemma addresses the case where a CCG of a quadratic MRS function is generated by a non-quadratic function.

Lemma 4.3.12. *Let f be an MRS function of n variable. Then there exists a quadratic MRS function h such that*

$$G_h = G_f$$

if and only if

$$f = x_1 x_d (OSANF)$$

for some $2 \leq d \leq n$ or some non-quadratic MRS function f such that

$$|DS(f)| = 1.$$

Proof. ($\Rightarrow$) Assume the conclusion is not true. Then, we have $|DS(f)| > 1$. Since $|DS(f)| > 1$ generates more than two edges at a vertex of G_f, there exists no quadratic MRS function h such that

$$G_h = G_f,$$

which is a contradiction.

($\Leftarrow$) If $f = x_1x_d(OSANF)$, the conclusion is immediate. If $f \neq x_1x_d(OSANF)$ and $DS(f) = \{k\}$ for $1 \leq k \leq \lceil \frac{n}{2} \rceil$, we can set

$$h = x_1x_k(OSANF).$$

□

Example 4.3.13. Let $n = 6$, and

$$\begin{aligned} f_1 &= x_1x_2(OSANF) \\ f_2 &= x_1x_2x_3x_4x_5x_6(OSANF) \\ h_1 &= x_1x_3(OSANF) \\ h_2 &= x_1x_3x_5(OSANF). \end{aligned}$$

Clearly, we have

$$|DS(f_1)| = |DS(f_2)| = |DS(h_1)| = |DS(h_2)| = 1,$$

$$G_{f_1} = G_{f_2}$$

and

$$G_{h_1} = G_{h_2}.$$

Lemma 4.3.14. *Let $f = x_1x_ix_k(OSANF)$ be a cubic MRS function of n variables. Let a, b, and c be distance elements of $x_1x_i(OSANF)$, $x_1x_{k-i+1}(OSANF)$, and $x_1x_k(OSANF)$, respectively. Then, the following statements are true:*

(1) If $a \neq b$, $a \neq c$, and $b \neq c$, then

$$G_f = G_{\|x_1x_i\|} \cup G_{\|x_1x_{k-i+1}\|} \cup G_{\|x_1x_k\|}.$$

(2) If $a \neq b$ and $b = c$, or $a \neq b$ and $a = c$, then

$$G_f = G_{\|x_1x_i\|} \cup G_{\|x_1x_{k-i+1}\|}.$$

(3) If $a = b$ and $b \neq c$, then

$$G_f = G_{\|x_1x_i\|} \cup G_{\|x_1x_k\|}.$$

(4) If $a = b = c$, then

$$G_f = G_{\|x_1x_i\|}.$$

Proof. For all instances, it is clear that

$$V(G_f) = V(G_{\|x_1x_i\|}) = V(G_{\|x_1x_{k-i+1}\|}) = V(G_{\|x_1x_k\|}).$$

So we focus on the equality of the edge sets.

(1) Since $a \neq b$, $a \neq c$, and $b \neq c$, an arbitrary vertex m has the edges

$$\{m,\, m \pm a \pmod{n}\}$$

$$\{m,\, m \pm b \pmod{n}\}$$

$$\{m,\, m \pm c \pmod{n}\}.$$

Also, each distance element generates a unique corresponding edge set. We have

$$\{\{j,\, j + a \pmod{n}\} | 1 \leq j \leq n\} = E(G_{\|x_1 x_i\|})$$

$$\{\{j,\, j + b \pmod{n}\} | 1 \leq j \leq n\} = E(G_{\|x_1 x_{k-i+1}\|})$$

$$\{\{j,\, j + c \pmod{n}\} | 1 \leq j \leq n\} = E(G_{\|x_1 x_k\|}).$$

Therefore,

$$E(G_f) = E(G_{\|x_1 x_i\|}) \cup E(G_{\|x_1 x_{k-i+1}\|}) \cup E(G_{\|x_1 x_k\|}),$$

and the claim holds.

(2) Since $a \neq b$ and $b = c$ (or $a \neq b$ and $a = c$), an arbitrary vertex m has the edges

$$\{m,\, m \pm a \pmod{n}\}$$

$$\{m,\, m \pm b \pmod{n}\},$$

Since the distance elements generate the following edges,

$$\{\{j,\, j+a \pmod n\}|1 \leq j \leq n\} = E(G_{\|x_1x_i\|})$$

$$\{\{j,\, j+b \pmod n\}|1 \leq j \leq n\} = E(G_{\|x_1x_{k-i+1}\|}).$$

Therefore,

$$E(G_f) = E(G_{\|x_1x_i\|}) \cup E(G_{\|x_1x_{k-i+1}\|}),$$

and the claim holds.

(3) The proof is similar to the one for (2).

(4) Since $a = b = c$, an arbitrary vertex m has the edges

$$\{m,\, m \pm a \pmod n\}.$$

The distance element generates the following edges

$$\{\{j,\, j+a \pmod n\}|1 \leq j \leq n\} = E(G_{\|x_1x_i\|}).$$

Therefore,

$$E(G_f) = E(G_{\|x_1x_i\|}),$$

and the claim holds. □

When we create another MRS by adding another variable, we can increase the cardinality of the distance set by at most two. Using this, we further generalize the idea of Lemma 4.3.14.

Lemma 4.3.15. *Let $f = x_1x_ix_j(OSANF)$ and $h = x_1x_ix_jx_k(OSANF)$ be MRS functions of n variable. Let a, b, and c be distance elements of $x_1x_j(OSANF)$, $x_1x_{k-j+1}(OSANF)$, and $x_1x_k(OSANF)$, respectively. Then, the following statements are true:*

(1) If $DS(h) = DS(f)$, then

$$G_h = G_f$$

(2) If $|DS(h)| = |DS(f)| + 1$, and a is a redundant distance element of f, then,

$$b = c$$

and

$$G_h = G_f \cup G_{\|x_1x_k\|}.$$

(3) If $|DS(h)| = |DS(f)| + 1$ and a is not a redundant distance element of f,

$$b \neq c$$

and

$$G_h = G_f \cup G_{\left\|x_1x_{k-j+1}\right\|} \cup G_{\|x_1x_k\|} - G_{\|x_1x_j\|}.$$

(4) If $|DS(h)| = |DS(f)| + 2$, a is a redundant distance element of f,

$$b \neq c,$$

and

$$G_h = G_f \cup G_{\left\|x_1 x_{k-j+1}\right\|} \cup G_{\|x_1 x_k\|} - G_{\|x_1 x_j\|}.$$

Proof. For all instances, the function h is obtained by removing the distance element a and adding the distance elements b and c. We can construct G_h from G_f, tracking the changes from $DS(f)$ to $DS(h)$. Clearly,

$$V(G_f) = V(G_h).$$

We also have a general construction of the edge set of G_h.

$$E(G_h) = E(G_f) \cup E(G_{\left\|x_1 x_{k-j+1}\right\|}) \cup E(G_{\|x_1 x_k\|}) - E(G_{\|x_1 x_j\|}).$$

(1) Since $DS(h) = DS(f)$, we have

$$E(G_f) = E(G_f) - E(G_{\|x_1 x_j\|}),$$

and

$$E(G_f) \supseteq E(G_{\left\|x_1 x_{k-j+1}\right\|}) \cup E(G_{\|x_1 x_k\|}).$$

Therefore,

$$E(G_f) = E(G_h).$$

(2) Since a is a redundant distance element of f,

$$E(G_f) = E(G_f) - E(G_{\|x_1 x_j\|}).$$

Since $|DS(h)| = |DS(f)| + 1$, $b = c$, or equivalently

$$E(G_{\|x_1x_{k-j+1}\|}) = E(G_{\|x_1x_k\|}).$$

Therefore,

$$G_h = G_f \cup G_{\|x_1x_k\|}.$$

(3) Since a is not a redundant distance element of f,

$$E(G_f) \supset E(G_f) - E(G_{\|x_1x_j\|}).$$

Additionally, $|DS(h)| = |DS(f)| + 1$. So, we have to have $b \neq c$. Therefore,

$$E(G_h) = E(G_f) - E(G_{\|x_1x_j\|}) \cup E(G_{\|x_1x_{k-j+1}\|}) \cup E(G_{\|x_1x_k\|}).$$

(4) If a is not a redundant distance element, or $b \neq c$, we have $DS(h) = DS(f)+1$ at most, which is a contradiction. Clearly,

$$E(G_h) = E(G_f) - E(G_{\|x_1x_j\|}) \cup E(G_{\|x_1x_{k-j+1}\|}) \cup E(G_{\|x_1x_k\|}),$$

and the claim follows. □

We extend Lemma 4.3.15 to the next theorem, whose proof is omitted, since it is somewhat similar.

Theorem 4.3.16. *Let $f = x_1x_{i2}x_{i3}\cdots x_{i(k-1)}x_{i_k}(OSANF)$ and $h = x_1x_{i2}x_{i3}\cdots x_{ik}x_{i(k+1)}$ $(OSANF)$ be MRS functions of n variables. Let a, b, and c be distance elements of $x_1x_{ik}(OSANF)$, $x_1x_{i(k+1)-ik+1}(OSANF)$, and $x_1x_{i(k+1)}(OSANF)$, respectively. Then, the following statements are true:*

(1) If $DS(h) = DS(f)$, then

$$G_h = G_f.$$

(2) If $|DS(h)| = |DS(f)| + 1$*, and* a *is a redundant distance element of* f*, then,*

$$b = c$$

and

$$G_h = G_f \cup G_{\|x_1 x_{i(k+1)}\|}.$$

(3) If $|DS(h)| = |DS(f)| + 1$ *and* a *is not a redundant distance element of* f*,*

$$b \neq c$$

and

$$G_h = G_f \cup G_{\|x_1 x_{i(k+1)-ik+1}\|} \cup G_{\|x_1 x_{i(k+1)}\|} - G_{\|x_1 x_{ik}\|}.$$

(4) If $|DS(h)| = |DS(f)| + 2$*, then* a *is a redundant distance element of* f*,*

$$b \neq c,$$

and

$$G_h = G_f \cup G_{\|x_1 x_{i(k+1)-ik+1}\|} \cup G_{\|x_1 x_{i(k+1)}\|} - G_{\|x_1 x_{ik}\|}.$$

The following theorems can be proved by fundamental number- and graph-theoretic techniques.

Theorem 4.3.17. *Let* f *be an MRS function of* n *variables. If* G_f *is disconnected, then* $1 \notin DS(f)$*, and every element in* $DS(f)$ *divides* n*.*

Proof. We prove this by contradiction. First, if $1 \in DS(f)$, G_f clearly contains the cycle $[1, 2, \ldots, n]$. Therefore, it is connected, which is a contradiction. Also, if there exists a distance element a of f such that $a \nmid n$, a is a generator of the group $\mathbb{Z}_n$ with respect to addition modulo n. And, we see that the following set of edges form C_n:

$$\{\{1, 1+a\}, \{1+a, 1+2a \bmod n\}, \ldots \{1+(n-1)a \bmod n, 1+na \bmod n\}\}$$

$$= \{\{1, 1+a\}, \{1+a, 1+2a \bmod n\}, \ldots \{1+(n-1)a \bmod n, 1\}\}.$$

This contradicts the fact that G_f is disconnected, since $C_n \in G_f$ implies G_f is connected.

□

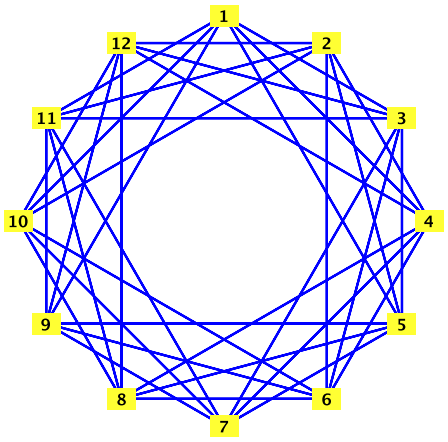


Figure 4.8: CCG of $f = x_1x_3x_6x_9(OSANF)$

The converse of the previous theorem does not hold, since there are instances where we can form a connected CCG with the nonzero distance elements that divide n. For example, let $n = 12$. Then, $f = x_1x_3x_6x_9(OSANF)$ has $1 \notin DS(f) = \{2, 3, 4\}$ and $2|12$, $3|12$ and $4|12$. However, G_f is connected, as seen on Figure 4.8. Next, we present a case where a CCG happens to be a complete graph.

Theorem 4.3.18. *Let f be an MRS function of n variables. Then, G_f is complete if and only if $DS(f) = \{1, 2, \ldots, \lfloor \frac{n}{2} \rfloor\}$.*

Proof. ($\Rightarrow$) Since G_f is regular, we make a case for the vertex 1. Since G_f is complete, vertex 1 is incident to the set of edges edges

$$\{\{1,2\},\{1,3\},\ldots,\{1,n\}\}.$$

By Definition 4.3.10,

$$\begin{aligned} DS(f) &= \{\min(2-1,n-1-2),\min(3-1,n-1-3),\ldots\min(n-1,n-n+1)\} \\ &= \{1,2,\ldots,\left\lfloor\frac{n}{2}\right\rfloor\}. \end{aligned}$$

($\Leftarrow$) By definition 4.3.10, the vertex 1 has a set of edges

$$\begin{aligned} &\{1,1\pm 1 \bmod n\},\{1,1\pm 2 \bmod n\},\ldots,\{1,1\pm\left\lfloor\frac{n}{2}\right\rfloor\} \\ =\ &\{\{1,1\},\{1,2\},\ldots,\{1,n\}\}. \end{aligned}$$

□

Corollary 4.3.19. *Let f be an MRS function of n variables. If $G_f = K_n$, then* $\deg(f) \geq \left\lfloor\frac{n}{2}\right\rfloor$.

Proof. By Theorem 4.3.18, $|DS(f)| = \left\lfloor\frac{n}{2}\right\rfloor$. Therefore, f needs at least $\left\lfloor\frac{n}{2}\right\rfloor$ variables in its OSANF. □

THIS PAGE INTENTIONALLY LEFT BLANK

5. TWO CONSTRUCTIONS OF BOOLEAN FUNCTIONS WITH GOOD CRYPTOGRAPHIC PROPERTIES

5.1. INTRODUCTION

The two key factors in designing cryptographic Boolean functions are security and speed. We achieve security by having good measures in as many cryptographic properties as possible for the Boolean functions in a cipher, such as balancedness to resist statistical attacks, high nonlinearity to address linear cryptanalysis, high algebraic degree against algebraic attacks, correlation immunity and resilience to deal with correlation attacks, and algebraic immunity to resist (fast) algebraic attacks. Speed is another important aspect, since we desire fast encryption and decryption. For example, the Carlet–Feng function has good cryptographic properties, but compared to other functions, it is not simple to generate or implement. This may cause certain ciphers to underperform. Security and speed often conflict with each other, since higher security usually implies slower speed. Here we present two constructions for good cryptographic Boolean functions, using a cryptographically strong base function, and three simple Boolean operations, namely affine transformation, concatenation, and complementation. One of the significant benefits from this construction is the flexibility to choose a base function with customizable cryptographic properties. We achieve security from the inherent qualities of the base function and obtain speed by the simple Boolean operations. In Chapter 6, we give applications for our constructions. This chapter is based on Chung, Stanica, Tan, and Wang [27].

5.2. CONSTRUCTION TECHNIQUES OF CRYPTOGRAPHIC BOOLEAN FUNCTIONS

In this section, we review fundamental construction techniques for cryptographic Boolean functions.

5.2.1. Concatenation

Given two base Boolean functions of f and g, both belonging to $\mathcal{B}_n$, we can construct another Boolean function, $h \in \mathcal{B}_{n+1}$, by concatenating their truth tables. We note that since the new function has to have 2^{n+1} elements in its truth table, the two functions concatenated must have the same number of variables or be the same length. To illustrate this point, if $h = f \parallel g$, $h \in \mathcal{B}_k$, $f \in \mathcal{B}_{k1}$, and $\mathcal{B}_{k2}$ with $k1, k2 \in \mathbb{N}$ and $k1 \neq k2$, we have $2^k = 2^{k1} + 2^{k2} = 2^{k1}(1 + 2^{k1-k2})$. This implies 2^k has an odd factor, which is a contradiction. Therefore, we provide the following preposition.

Proposition 5.2.1. *Let f and g be two Boolean functions. If $h = f \parallel g$, then f and g have the same number of variables.*

Concatenating two Boolean functions introduces a new variable to the ANF of the concatenated function. The following useful lemma illustrates how we can obtain the ANF of the new function from the ANFs of the base functions.

Lemma 5.2.2. *Let $f, g \in B_{n-1}$. If $h = f \parallel g$ with $h \in B_n$, then*

$$h(\mathbf{x}) = (x_n \oplus 1)f(\mathbf{x}_{n-1}) \oplus x_n g(\mathbf{x}_{n-1}),$$

where $\mathbf{x}_{n-1} = (x_1, x_2, \ldots, x_{n-1})$ and $\mathbf{x} = (x_1, x_2, \ldots, x_n)$.

Example 5.2.3. We illustrate Lemma 5.2.2 with two functions f and g on Table 5.1. We can convert the truth tables to ANFs as below.

$$f(\mathbf{x}) = x_1 \oplus x_2 \oplus x_3 \oplus x_1x_3 \oplus x_2x_3 \oplus x_1x_2x_3$$

$$g(\mathbf{x}) = 1 \oplus x_1 \oplus x_3 \oplus x_1x_2 \oplus x_2x_3 \oplus x_1x_2x_3$$

We confirm the following equation of ANFs of the functions using Lemma 5.2.2 and Table 5.2.

x_3	x_2	x_1	$f(\mathbf{x})$	$g(\mathbf{x})$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

Table 5.1: Truth Table of f and g

$$h(\mathbf{x}) = (x_4 \oplus 1)f(\mathbf{x}_{n-1}) \oplus x_4 g(\mathbf{x}_{n-1})$$

$$= x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_1x_3 \oplus x_2x_3 \oplus x_2x_4 \oplus x_1x_2x_3 \oplus x_1x_2x_4 \oplus x_1x_3x_4$$

x_4	x_3	x_2	x_1	$h(\mathbf{x})$	x_4	x_3	x_2	x_1	$h(\mathbf{x})$
0	0	0	0	0	1	0	0	0	1
0	0	0	1	1	1	0	0	1	0
0	0	1	0	1	1	0	1	0	1
0	0	1	1	0	1	0	1	1	1
0	1	0	0	1	1	1	0	0	0
0	1	0	1	1	1	1	0	1	1
0	1	1	0	1	1	1	1	0	1
0	1	1	1	0	1	1	1	1	0

Table 5.2: Truth Table of $h = f \parallel g$

The following theorem by Siegenthaler shows that a technique as simple as concatenation can be used to preserve certain cryptographic properties.

Theorem 5.2.4. [23] *If Boolean functions f, $g \in B_n$ have correlation immunity of order k, then $h = f \parallel g$ has correlation immunity of order k.*

5.2.2. Kronecker Product

The Kronecker product is a matrix operation that takes two matrices of arbitrary size and outputs a block matrix.

Definition 5.2.5. Given $A = \{a_{ij}\}$, an $m \times n$ matrix and $B = \{b_{rs}\}$, a $p \times q$ matrix. The Kronecker product of A and B, denoted by $A \otimes B$ is an $mp \times nq$ matrix,

$$\begin{aligned} A \otimes B &= \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix} \\ &= \begin{bmatrix} a_{11}b_{11} & \cdots & a_{1n}b_{1q} \\ \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & \cdots & a_{mn}b_{pq} \end{bmatrix}. \end{aligned}$$

The Kronecker product can be used to generate a higher-dimensional bent functions from a base bent function.

Theorem 5.2.6. [67] *Let a $4k$-dimensional column vector $\mathbf{x}$ represent the truth table of a bent function with $k = 1, 2, \ldots$. Then,*

$$\mathbf{z} = \mathbf{x} \otimes \mathbf{x}$$

is a bent function expressed in a $16k^2$-dimensional column vector.

In another example, the Kronecker product is a key concept to prove the following theorem, which addresses a construction of bent function.

Theorem 5.2.7. [67] *Let two Boolean functions f and g such that $f : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$ and $g : \mathbb{F}_2^m \longrightarrow \mathbb{F}_2$. Then the Boolean function $h : \mathbb{F}_2^{n+m} \longrightarrow \mathbb{F}_2$, defined by $h(\mathbf{z}) = f(\mathbf{x}) \oplus g(\mathbf{y})$ with $\mathbf{z} = \mathbf{x} \| \mathbf{y}$ is bent if and only if f and g are bent.*

This theorem shows how a Boolean function of $2k$ variables, $f(\mathbf{x}) = x_1x_2 \oplus x_3x_4 \oplus \cdots \oplus x_{2k-1}x_{2k}$ with k ≥ 1 is bent. The direct-sum method is a key component of various bent function constructions including the constructions of Maiorana and McFarland [68], [69], and Carlet [70], [71], and Canteaut et al. [30].

5.2.3. Affine Operations

We can integrate various operations that are conceptually linear to a construction method to have significant effects. For example, linear transformation of variables, complemetation of domain or function values, and adding polynomials are frequently used for construction and analysis.

Example 5.2.8. If a Boolean function f is bent, then $f \oplus l$ is bent for any affine function l [4, p. 83]. Let A be an $n \times n$ invertible matrix over $\mathbb{F}_2$ and $\mathbf{v} \in \mathbb{F}_2^n$. If a Boolean function f of n variables is bent, then $g(\mathbf{x}) = f(A\mathbf{x} \oplus \mathbf{v})$ is bent [4, p. 84]. Therefore, $h(\mathbf{x}) = f(A\mathbf{x} \oplus \mathbf{v}) \oplus l$ is bent as well.

5.3. TWO CONSTRUCTIONS TO ADDRESS SECURITY AND SPEED

We introduce two constructions [27] based on functions $f_i \in \mathcal{B}_{n-2}$ where $i = 1, 2, \ldots$.

Construction 1.

For $\{i, j\} = \{1, 2\}$, we define the functions on $\mathbb{F}_2^n$:

$$f_i \parallel f_j \parallel f_i \parallel \bar{f}_j; \;\; f_i \parallel f_j \parallel \bar{f}_i \parallel f_j; \;\; f_i \parallel \bar{f}_j \parallel f_i \parallel f_j; \;\; \bar{f}_i \parallel f_j \parallel f_i \parallel f_j;$$

$$f_i \parallel f_j \parallel f_j \parallel \bar{f}_i; \;\; f_i \parallel f_j \parallel \bar{f}_j \parallel f_i; \;\; f_i \parallel \bar{f}_j \parallel f_j \parallel f_i; \;\; \bar{f}_i \parallel f_j \parallel f_j \parallel f_i.$$

Construction 2.

For $\{i, j\} = \{1, 2\}$, we define the functions on $\mathbb{F}_2^n$:

$$f_i \parallel f_j \parallel \bar{f}_i \parallel \bar{f}_j;\ \ f_i \parallel f_j \parallel \bar{f}_j \parallel \bar{f}_i;\ \ \bar{f}_i \parallel \bar{f}_j \parallel f_i \parallel f_j;\ \ \bar{f}_i \parallel \bar{f}_j \parallel f_j \parallel f_i.$$

We observe that some functions in the constructions are affine equivalent to each other. For example, given two functions u and v of $n-1$ variables with $\mathbf{x} \in \mathbb{F}_2^n$,

$$\begin{aligned}(u \parallel \bar{v})(\mathbf{x}) &= (x_n \oplus 1)u \oplus x_n(v \oplus 1) \\ &= (x_n \oplus 1)u \oplus x_n v \oplus x_n \\ &= (u \parallel v)(\mathbf{x}) \oplus x_n\end{aligned}$$

by Definition 3.2.1. Therefore,

$$u \parallel v \sim u \parallel \bar{v}.$$

Also,

$$(u \parallel v)(\mathbf{x}) = ((v||u)(\mathbf{x} \oplus (0, \ldots, 0, 1))$$

due to the lexicographical order of domain. So we have

$$u \parallel v \sim v \parallel u,$$

where $\sim$ signifies affine equivalence.

By setting $u = f_i \parallel f_j$ and $v = f_i \parallel \bar{f}_j$, it is clear that $u \parallel v = f_i \parallel f_j \parallel f_i \parallel \bar{f}_j$ is affine equivalent to $u \parallel \bar{v} = f_i \parallel f_j \parallel \bar{f}_i \parallel f_j$. By similar arguments, we have for Construction 1,

$$f_i \parallel f_j \parallel f_i \parallel \bar{f}_j \sim \begin{cases} f_i \parallel f_j \parallel \bar{f}_i \parallel f_j \\ f_i \parallel \bar{f}_j \parallel f_i \parallel f_j \\ \bar{f}_i \parallel f_j \parallel f_i \parallel f_j \end{cases}$$

and

$$f_i \parallel f_j \parallel f_j \parallel \bar{f}_i \sim \begin{cases} f_i \parallel f_j \parallel \bar{f}_j \parallel f_i \\ f_i \parallel \bar{f}_j \parallel f_j \parallel f_i \\ \bar{f}_i \parallel f_j \parallel f_j \parallel f_i \end{cases}.$$

For Construction 2,

$$f_i \parallel f_j \parallel \bar{f}_i \parallel \bar{f}_j = \bar{f}_i \parallel \bar{f}_j \parallel f_i \parallel f_j \oplus 1$$

and

$$f_i \parallel f_j \parallel \bar{f}_j \parallel \bar{f}_i \;\; = \;\; \bar{f}_i \parallel \bar{f}_j \parallel f_j \parallel f_i \oplus 1,$$

Therefore, we have

$$f_i \parallel f_j \parallel \bar{f}_i \parallel \bar{f}_j \;\; \sim \;\; \bar{f}_i \parallel \bar{f}_j \parallel f_i \parallel f_j$$

and

$$f_i \parallel f_j \parallel \bar{f}_j \parallel \bar{f}_i \;\; \sim \;\; \bar{f}_i \parallel \bar{f}_j \parallel f_j \parallel f_i.$$

There have been some constructions which use some components of our constructions. For example, the bentness, the resiliency, and the normality properties of concatenated bent functions were considered in [72, 73]. The normality of $f_1 \parallel f_2 \parallel f_2 \parallel \bar{f}_1$ for arbitrary function f_i with $i = 1, 2$ is mentioned in [74]. Our constructions address the instance where f_i's are affine equivalent to each other, and we cover other configurations. Moreover, we explore more than the normality of the functions. $f \in \mathcal{B}_n$ satisfies the *high degree product* (HDP) of order n if, for any non-annihilating function g of degree $1 \leq e \leq \lceil n/2 \rceil - 1$, the degree $d = \deg(gf)$ satisfies $e + d \geq n$ [75]. In [75], Pasalic introduced a concatenation of four functions which requires each function to have maximum algebraic immunity, to show that the notion of HDP can measure resistance to fast algebraic attacks.

Remark 5.3.1. In [76], Wang et al. demonstrated that the construction based on a four-function concatenation in [75] does not always produce HDP function.

5.4. CRYPTOGRAPHIC PROPERTIES OF THE TWO CONSTRUCTIONS

We start with algebraic immunity and nonlinearity. To set the stage for these properties, we take a look at the Walsh-Hadamard transform of the functions. The relationship between Walsh-Hadamard transform and the function formed by concatenating two or four functions of the same variables are well known. We generalize the relationship and present the next lemma, which describes the Walsh-Hadamard coefficients of g (in some dimension) to the Walsh-Hadamard coefficients of its 2^{-k} ($k \geq 1$) concatenated parts.

Lemma 5.4.1. [27] *If* $g(\mathbf{x}, x_{n+1}, \ldots, x_{n+r}) = f_1(\mathbf{x}) \| f_2(\mathbf{x}) \| \cdots \| f_{2^r}(x) = \mathop{\Vert}_{i=1}^{2^r} f_i(\mathbf{x})$*, then*

$$
\begin{aligned}
& W_g(\mathbf{u}, u_{n+1}, \ldots, u_{n+r}) \\
& = W_{f_1}(\mathbf{u}) + (-1)^{u_{n+1}} W_{f_2}(\mathbf{u}) + \cdots + (-1)^{u_{n+1}+\cdots+u_{n+r}} W_{f_{2^r}}(\mathbf{u}) \\
& = \sum_{k=1}^{2^r} (-1)^{\mathbf{a}(k)\cdot\mathbf{u}'} W_{f_k}(\mathbf{u}),
\end{aligned}
$$

where $r \in \mathbb{N}$, $\mathbf{a}(k)$ *is the* k*th lexicographically ordered vector in* $\mathbb{F}_2^r$*, and* $\mathbf{u}' = (u_{n+1}, \ldots, u_{n+r})$.

Proof. We show our result by induction on r. If $r = 1$,

$$
\begin{aligned}
W_g(\mathbf{u}, u_{n+1}) &= \sum_{(\mathbf{x}, x_{n+1}) \in \mathbb{F}_2^{n+1}} (-1)^{g(\mathbf{x}, x_{n+1}) + \mathbf{u}\cdot\mathbf{x} + u_{n+1} x_{n+1}} \\
&= \sum_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{g_1(\mathbf{x}) + \mathbf{u}\cdot\mathbf{x}} + (-1)^{u_{n+1}} \sum_{\mathbf{x} \in \mathbb{F}_2^n} (-1)^{g_2(\mathbf{x}) + \mathbf{u}\cdot\mathbf{x}} \\
&= W_{g_1}(\mathbf{u}) + (-1)^{u_{n+1}} W_{g_2}(\mathbf{u}).
\end{aligned}
$$

For the induction hypothesis, we assume,

$$W_g(\mathbf{u}, u_{n+1}, \ldots, u_{n+r}) = \sum_{k=1}^{2^r} (-1)^{\mathbf{a}(k)\cdot\mathbf{u}'} W_{f_k}(\mathbf{u}),$$

for

$$g''(\mathbf{x}, x_{n+1}, \ldots, x_{n+r+1}) = f_1(\mathbf{x}) \| f_2(\mathbf{x}) \| \cdots \| f_{2^{r+1}}(\mathbf{x}) = g \| g' = \mathop{\Vert}_{i=1}^{2^{r+1}} f_i(\mathbf{x}),$$

where $g' = f_{2^r+1}(\mathbf{x}) \| f_{2^r+2}(\mathbf{x}) \| \cdots \| f_{2^{r+1}}(\mathbf{x})$.

Then, we have

$$\begin{aligned}
& W_{g''}(\mathbf{u}, u_{n+1}, \ldots, u_{n+r+1}) \\
= \;& W_g(\mathbf{u}, u_{n+1}, \ldots, u_{n+r}) + (-1)^{u_{n+r+1}} W_{g'}(\mathbf{u}, u_{n+1}, \ldots, u_{n+r}) \\
= \;& W_{f_1}(\mathbf{u}) + (-1)^{u_{n+1}} W_{f_2}(\mathbf{u}) + \cdots + (-1)^{u_{n+1}+\cdots+u_{n+r+1}} W_{f_{2^{r+1}}}(\mathbf{u}) \\
= \;& \sum_{k=1}^{2^{r+1}} (-1)^{\mathbf{a}(k)\cdot\mathbf{u}'} W_{f_k}(\mathbf{u}),
\end{aligned}$$

which shows our result. □

The next lemma shows what happens to algebraic immunity when XORing two functions.

Lemma 5.4.2. [77, Lemma 1] *For any $f \in \mathcal{B}_n$ and any $l \in \mathcal{A}_n$,*

$$AI(f) - 1 \leq AI(f \oplus l) \leq AI(f) + 1.$$

In general, for any $f \in \mathcal{B}_n$ and any function $h \in \mathcal{B}_n$ with $\deg(h) = k$,

$$AI(f) - k \leq AI(f \oplus h) \leq AI(f) + k.$$

The next lemma shows how algebraic immunity behaves when concatenating two functions.

Lemma 5.4.3. [77, Proposition 1] *Let g_1, g_2 be two Boolean functions in the variables $x_1, \ldots, x_n$ with $AI(g_1) = d_1, AI(g_2) = d_2$, and let $g = (1 \oplus x_{n+1})g_1 \oplus x_{n+1}g_2 \in \mathcal{B}_{n+1}$. Then, the following hold:*

If $d_1 \neq d_2$, then $AI(g) = \min\{d_1, d_2\} + 1$.

If $d_1 = d_2 (=: d)$, then $d \leq AI(g) \leq d + 1$. Further, $AI(g) = d$ if and only if there exists $f_1, f_2 \in \mathcal{B}_n$ of algebraic degrees d that either both annihilate g_1, g_2, or both annihilate $\bar{g}_1, \bar{g}_2$, and $\deg(f_1 \oplus f_2) \leq d - 1$.

For our next result, we let $f_1 \in \mathcal{B}_{n-2}$ in Construction 1 and 2 be any balanced function and $f_2(\mathbf{x}) = f_1(A\mathbf{x} \oplus \mathbf{b})$, where A is an $(n-2)$ by $(n-2)$ invertible matrix over $\mathbb{F}_2$ and $\mathbf{b}$ is an $(n-2)$ dimensional vector over $\mathbb{F}_2$. We note that, since f_1 and f_2 are affine equivalent, we have $\deg(f_1) = \deg(f_2)$, $AI(f_1) = AI(f_2)$ and $nl(f_1) = nl(f_2)$.

Theorem 5.4.4. [27] *Let $f \in \mathcal{B}_n$ be given by Constructions 1 or 2. $f_1, f_2 \in \mathcal{B}_{n-2}$ are nonconstant and affine equivalent. Then, f is balanced.*

$$\deg(f) = \max\{\deg(f_1), \deg(f_1 \oplus f_2) + 1\},$$

and

$$AI(f) \geq \min\{AI(f_1||f_2), AI(f_1||\bar{f}_2)\} \geq AI(f_1).$$

Moreover,

$$nl(f) = 2^{n-2} + 2nl(f_1),$$

for functions in Construction 1, and

$$nl(f) = 4nl(f_1),$$

for functions in Construction 2.

Proof. We prove the result for Constuction 1 for two cases, since the others are similar. First, let $f = f_1||f_2||f_1||\bar{f}_2$. We observe that

$$\begin{aligned} f &= (x_n \oplus 1)(f_1 \parallel f_2) \oplus x_n(f_1 \parallel \bar{f}_2) \\ &= (x_n \oplus 1)((x_{n-1} \oplus 1)f_1 \oplus x_{n-1}f_2) \\ &\quad \oplus x_n((x_{n-1} \oplus 1)f_1 \oplus x_{n-1}(f_2 \oplus 1)) \\ &= x_{n-1}f_1 \oplus f_1 \oplus x_{n-1}f_2 \oplus x_n x_{n-1} \\ &= (f_1 \parallel f_2) \oplus x_n x_{n-1}. \end{aligned}$$

Since f_1 and f_2 are nonconstant,

$$\begin{aligned} \deg(f) &= \deg(f_1||f_2) \\ &= \max\{\deg(f_1), \deg(f_1 \oplus f_2) + 1\}. \end{aligned}$$

Since

$$(f_1||\bar{f}_2)(\mathbf{x}_{n-1}) = (f_1||f_2)(\mathbf{x}_{n-1}) \oplus x_{n-1},$$

where $\mathbf{x}_{n-1} = (x_1, x_2, \ldots, x_{n-1})$, by Lemma 5.4.2,

$$|AI(f_1||f_2) - AI(f_1||\bar{f}_2)| \leq 1.$$

So, we check two possibilities.

If $AI(f_1||f_2) = AI(f_1||\bar{f}_2)$, by Lemma 5.4.3

$$AI(f) \geq AI(f_1||f_2) \geq AI(f_1).$$

If $|AI(f_1||f_2) - AI(f_1||\bar{f}_2)| = 1$, then Lemma 5.4.3 shows that

$$AI(f) = \min\{d, d+1\} + 1 = d + 1,$$

where $\min\{AI(f_1||f_2), AI(f_1||\bar{f}_2)\} = d$.

Second, let $f = f_1||f_2||f_2||\bar{f}_1$. Then,

$$\begin{aligned} f &= (x_n \oplus 1)(f_1 \parallel f_2) \oplus x_n(f_2 \parallel \bar{f}_1) \\ &= x_{n-1}f_1 \oplus f_1 \oplus x_{n-1}f_2 \oplus x_n f_1 \oplus x_n f_2 \oplus x_n x_{n-1} \\ &= (f_1 \parallel f_2) \oplus x_n(f_1 \oplus f_2 \oplus x_{n-1}). \end{aligned}$$

So, we have

$$\begin{aligned} \deg(f) &= \deg(f_1||f_2) \\ &= \max\{\deg(f_1),\, \deg(f_1 \oplus f_2) + 1\}. \end{aligned}$$

The algebraic immunity computation does not change in this case.

To find the nonlinearity, we consider only $f = f_1||f_2||f_1||\bar{f}_2$ of Construction1 since the proofs for the other cases are similar. Using Lemma 5.4.1, we obtain

$$\begin{aligned} W_f(\mathbf{u}, u_{n-1}, u_n) &= W_{f_1}(\boldsymbol{u}) + (-1)^{u_{n-1}} W_{f_2}(\mathbf{u}) \\ &\quad +(-1)^{u_n} W_{f_1}(\mathbf{u}) + (-1)^{u_{n-1}+u_n} W_{\bar{f}_2}(\mathbf{u}) \\ &= (1 + (-1)^{u_n})\, W_{f_1}(\mathbf{u}) + (-1)^{u_{n-1}} (1 - (-1)^{u_n})\, W_{f_2}(\mathbf{u}). \end{aligned}$$

Thus, $W_f(\mathbf{u}, u_{n-1}, 0) = 2W_{f_1}(\mathbf{u})$ and $W_f(\mathbf{u}, u_{n-1}, 1) = 2(-1)^{u_{n-1}} W_{f_2}(\mathbf{u})$. It follows that

$$\max_{(\mathbf{u},u_{n-1},u_n)\in\mathbb{F}_2^n} |W_f(\mathbf{u}, u_{n-1}, u_n)| = 2 \max_{\mathbf{u}\in\mathbb{F}_2^{n-2}} |W_{f_1}(\mathbf{u})| = 2^{n-1} - 4nl(f_1)).$$

Therefore,

$$nl(f) = 2^{n-2} + 2nl(f_1).$$

Next, we take two cases of Construction 2, as they are slightly different. The other cases are similar to these.

Case 1. Let $f = f_1||f_2||\bar{f}_1||\bar{f}_2$. As above,

$$\begin{aligned} W_f(\mathbf{u}, u_{n-1}, u_n) &= W_{f_1}(\mathbf{u}) + (-1)^{u_{n-1}} W_{f_2}(\mathbf{u}) \\ &\quad +(-1)^{u_n} W_{\bar{f}_1}(\mathbf{u}) + (-1)^{u_{n-1}+u_n} W_{\bar{f}_2}(\mathbf{u}) \\ &= (1 - (-1)^{u_n})\, W_{f_1}(\mathbf{u}) + (-1)^{u_{n-1}} (1 - (-1)^{u_n})\, W_{f_2}(\mathbf{u}) \\ &= (1 - (-1)^{u_n}) \left(W_{f_1}(\mathbf{u}) + (-1)^{u_{n-1}} W_{f_2}(\mathbf{u})\right). \end{aligned}$$

Case 2. Let $f = f_1||f_2||\bar{f}_2||\bar{f}_1$. Then,

$$\begin{aligned} W_f(\mathbf{u}, u_{n-1}, u_n) &= W_{f_1}(\mathbf{u}) + (-1)^{u_{n-1}} W_{f_2}(\mathbf{u}) \\ &\quad + (-1)^{u_n} W_{\bar{f}_2}(\mathbf{u}) + (-1)^{u_{n-1}+u_n} W_{\bar{f}_1}(\mathbf{u}) \\ &= \left(1 - (-1)^{u_{n-1}+u_n}\right) W_{f_1}(\mathbf{u}) + (-1)^{u_{n-1}} \left(1 - (-1)^{u_{n-1}+u_n}\right) W_{f_2}(\mathbf{u}) \\ &= \left(1 - (-1)^{u_n+u_{n-1}}\right) \left(W_{f_1}(\mathbf{u}) + (-1)^{u_{n-1}} W_{f_2}(\mathbf{u})\right). \end{aligned}$$

Regardless of the case, we see that for Construction 2, we have

$$\begin{aligned} \max_{(\mathbf{u},u_{n-1},u_n)\in\mathbb{F}_2^n} |W_f(\mathbf{u}, u_{n-1}, u_n)| &= 4 \max_{\mathbf{u}\in\mathbb{F}_2^{n-2}} |W_{f_1}(\mathbf{u})| \\ &= 2^n - 8nl(f_1)), \end{aligned}$$

which renders

$$nl(f) = 4nl(f_1).$$

□

We note that the nonlinearity in Construction 1 is much better than that of Construction 2 with $n \geq 3$. It is attributed to the following reasoning. Since $f_1 \in \mathcal{B}_{n-2}$,

$$nl(f_1) \leq 2^{n-3} - 2^{n/2-2} < 2^{n-3}.$$

Therefore,

$$nl(f) = 2^{n-2} + 2nl(f_1) > 4nl(f_1).$$

As for the algebraic immunity, in most cases, $\deg(f_1(\mathbf{x}A \oplus \mathbf{b}) \oplus f_1) = \deg(f_1)$. That is, $\deg(f) = \deg(f_1) + 1$. By Lemma 5.4.3, it is usually the case that

$$AI(f_1||f_2) = AI(f_1) + 1.$$

That is,

$$AI(f) \geq AI(f_1) + 1.$$

Also, we note $nl(f)$ is much better than $nl(f_1)$. Additionally, the fast correlation attack on f has an on-line complexity proportional to $\left(\frac{1}{\epsilon}\right)^2$ where $\epsilon(f) = \frac{nl(f)}{2^n} - \frac{1}{2}$ is the the bias of nonlinearity [20]. The bias for Construction 1 is

$$\begin{aligned}
\epsilon(f) &= \frac{nl(f)}{2^n} - \frac{1}{2} \\
&= \frac{1}{4} - \frac{nl(f_1)}{2^{n-1}} - \frac{1}{2} \\
&= \frac{1}{2}\left(\frac{nl(f_1)}{2^{n-2}} - \frac{1}{2}\right).
\end{aligned}$$

This shows our constructions improve against correlation attacks when compared to the base function.

Proposition 5.4.5. [75, Proposition 1] *Let $f = f_1 \parallel f_2 \parallel f_3 \parallel f_4$ be an element of $\mathcal{B}_{n+2}$ where n is even. Let $f_i \in \mathcal{B}_n$ with $i = 1, \ldots, 4$ have maximum algebraic immunity, that is $AI(f_i) = \left\lceil \frac{n}{2} \right\rceil$. Let f_1 be such that for any function g of $\deg(g) = e$, $e \in \left[1, \left\lceil \frac{n}{2} \right\rceil - 1\right]$, we have $\deg(f_1 g) = d \geq AI(f_1)$, and $e + d \geq n$. Also let $f_1 = f_3 \oplus 1$. Then*

$$AI(f) = \left\lceil \frac{n}{2} \right\rceil + 1,$$

which shows that f has maximum algebraic immunity.

Using Proposition 5.4.5, we can further infer that if we take $f_1,\ f_2 \in \mathcal{B}_n$ with n even of maximum AI, with the property that for any function g of algebraic degree $1 \leq e \leq \left\lceil \frac{n}{2} \right\rceil - 1$, we have $\deg(f_1 g) = d \geq AI(f_1)$ and $e + d \geq n$, then $f = f_1||f_2||\bar{f_1}||f_2$ has maximum AI. The Boolean functions with maximum algebraic immunity are called *perfect algebraic immune* (PAI) [78]. Liu et al. introduced the notion of PAI and showed that if f_1 is a balanced PAI, then $n = 2^k + 1$ for some k; if f_1 is unbalanced, then $n = 2^k$, for some k [78, Theorem 7]. Next, we present the results related to normality of our constructions.

Theorem 5.4.6. [27] *Let $f_i,\ f_j \in \mathcal{B}_{n-2}$. If f_i or f_j, whichever does not have its complementation in Construction 1, is k-normal, then the functions f of Construction 1 are at least $(k+1)$-normal.*

Proof. Due to the affine equivalence to $f_i,\ f_j$ is k-normal. If f_i is invariant, say 0 on a k-dimensional flat, then $\bar{f_i}$ is invariant with 1 on the same flat, which shows that $\bar{f_i}$ is k-normal. We prove for the case $f = f_i \| f_j \| f_i \| \bar{f_j}$ only, since the others can be shown by similar arguments. We show the existence of a $(k+1)$-dimensional affine subspace where $f(\mathbf{x})$ is a constant. Let $\mathbf{z}_1, \ldots \mathbf{z}_k \in$ be k distinct, linearly independent vectors in $\mathbb{F}_2^{n-2}$, $\mathbf{d} = (d_1, d_2, \ldots, d_{n-2})$ be a vector in $\mathbb{F}_2^{n-2}$, and $a_i \in \mathbb{F}_2$ be for $1 \leq i \leq k$. We define a k-dimensional flat $G = \{\mathbf{x} \in \mathbb{F}_2^{n-2} \mid \mathbf{x} = a_1\mathbf{z}_1 + a_2\mathbf{z}_2 + \cdots + a_k\mathbf{z}_k + \mathbf{d},\ a_i = \mathbb{F}_2,\ 1 \leq i \leq k\}$ such that $f_i|_G = 0$. In construction of f, we integrate two variables, x_{n-1} and x_n into the domain of f_i, and we can construct a $(k+1)$-dimensional flat in the following way. Let $\mathbf{z}_l = (z_{l1}, z_{l2}, \ldots, z_{l(n-2)})$ where $1 \leq l \leq k$. We set

$$\mathbf{z}'_l = (z_{l1}, z_{l2}, \ldots, z_{l(n-2)}, 0, 0),$$

$$\mathbf{z}'_{k+1} = (0, \ldots, 0, 1),$$

and

$$\mathbf{d}' = (d_1, d_2, \ldots, d_{n-2}, 0, 0)$$

where $\mathbf{z}'_{k+1},\ \mathbf{d}' \in \mathbb{F}_2^n$. Then

$$G' = \{\mathbf{x}' \in \mathbb{F}_2^n \mid \mathbf{x}' = a_1\mathbf{z}'_1 + a_2\mathbf{z}'_2 + \cdots + a_{k+1}\mathbf{z}'_{k+1} + \mathbf{d}',\ a_i = \mathbb{F}_2,\ 1 \leq i \leq k+1\}.$$

If a vector $\mathbf{x}' \in G'$ with $a_{k+1} = 0$, then f follows the first f_i in the construction. If a vector $\mathbf{x}' \in G'$ with $a_{k+1} = 1$, then f follows the third f_i in the construction. Therefore, G' is a $(k+1)$-dimensional flat such that $f\,|_{G'} = 0$. □

Generally, it is difficult to establish a proper limit to the normality of a function. Let f_i or f_j, whichever does not have its complementation in Construction 1, be k-normal but not $k+1$-normal, and we show that the function f of Construction 1 cannot have a constant function value on the $k+2$-dimensional flat $H = \{a_1\mathbf{e}_{i_1} \oplus \cdots \oplus a_{k+2}\mathbf{e}_{i_{k+2}} \oplus \mathbf{d}\}$, where $\mathbf{d} = (y_1, \ldots y_n)$ is a fixed vector in $\mathbb{F}_2^n$ and $\mathbf{e}_{i_m} = (x_1, \ldots, x_n)$ is an elementary vector such that $x_j = 1$ if and only if $j = i_m$ with $1 \leq i_m \leq n$. We assume $f = f_i\|f_j\|f_i\|\bar{f}_j$ since the others can be shown by similar arguments. Let us also assume that H exists. We observe that y_{i_m} is irrelevant (whether it is 0 or 1) due to $\mathbf{e}_{i_m}$, so we set $\mathbf{d}$ with $y_{i_1} = \ldots = y_{i_{k+2}} = 0$. To illustrate better, we rewrite the restriction of our function to H as follows:

$$\begin{aligned}
f(\mathbf{x})|_H &= (\bar{x}_{n-1}f_i \oplus x_{n-1}f_j)\|(\bar{x}_{n-1}f_i \oplus x_{n-1}\bar{f}_j)\,|_H \\
&= \bar{x}_n(\bar{x}_{n-1}f_i \oplus x_{n-1}f_j) \oplus x_n(\bar{x}_{n-1}f_i \oplus x_{n-1}\bar{f}_j)\,|_H \\
&= \bar{x}_{n-1}(\bar{x}_n f_i \oplus x_n f_i) \oplus x_{n-1}(\bar{x}_n f_j \oplus x_n \bar{f}_j)\,|_H \\
&= f_i \oplus x_{n-1}f_i \oplus x_{n-1}f_j \oplus x_{n-1}x_n\,|_H.
\end{aligned}$$

Without loss of generality, we assume $f(\mathbf{x}) = 0$ for all $\mathbf{x} = (x_1, \ldots, x_n) \in H$, and we examine the following cases, depending upon the values of x_{n-1} and x_n.

Case 1: $n-1,\ n \notin \{i_1, i_2, \ldots, i_{k+2}\}$. Then $x_{n-1} = d_{n-1}$, and $d_n = x_n$. We observe that for all possible values for x_{n-1} and x_n, $f\,|_H$ is one of the functions, f_i, f_j, or $\bar{f}_j$. Since each function is only k-normal, there exists at least one $\mathbf{x} \in H$ such that $f(\mathbf{x})|_H = 1$, which is a contradiction.

Case 2: $n-1 \notin \{i_1, i_2, \ldots, i_{k+2}\}$ and $x_n \in \{i_1, i_2, \ldots, i_{k+2}\}$. Then $x_{n-1} = d_{n-1}$. If $x_{n-1} = 0$, then regardless of the value of x_n, $f\,|_H$ follows the function, f_i. We note that we can only increase the normality to $k+1$ using x_n, since f_i is k-normal. Therefore, there exists at least one $\mathbf{x} \in H$ such that $f(\mathbf{x})|_H = 1$, which is a contradiction. If $x_{n-1} = 1$, $f\,|_H$ follows the function, f_j with $x_n = 0$ or $\bar{f}_j$ with $x_n = 1$. Clearly, $f\,|_H$ is at most k-normal, since $\bar{f}_j = f_j \oplus 1$. So, there exists at least one $\mathbf{x} \in H$ such that $f(\mathbf{x})|_H = 1$, which is a contradiction.

Case 3: $n \notin \{i_1, i_2, \ldots, i_{k+2}\}$ and $x_{n-1} \in \{i_1, i_2, \ldots, i_{k+2}\}$. Then $d_n = x_n$. If $x_n = 0$, then $f\,|_H$ follows the function, $f_i \| f_j$. Also, if $x_n = 1$, then $f\,|_H$ follows the function, $f_i \| \bar{f}_j$. In both instances, we can only increase the normality to $k+1$, since f_i, f_j and $\bar{f}_j$ are k-normal.

Case 4: $x_{n-1},\ x_n \in \{i_1, i_2, \ldots, i_{k+2}\}$. In this case $f\,|_H$ follows $f_i \| f_j \| f_i \| \bar{f}_j\,|_H$, and any two vectors $\mathbf{x}', \mathbf{x}'' \in H$ in the forms of $\mathbf{x}' = (a_1, \ldots, a_{n-2}, 1, 0)$ and $\mathbf{x}'' = (b_1, \ldots, b_{n-2}, 1, 1)$ with $a_i,\ b_i \in \mathbb{F}_2,\ 1 \leq i \leq n-2$ have opposite function values. Therefore, we have a contradiction.

Under what conditions the functions of Construction 1 is $k+2$-normal remains an open problem. Using a similar approach, we can show a similar result for the functions of Construction 2.

Theorem 5.4.7. [27] *If f_i is k-normal, then the functions f of Construction 2 are k or $k+1$-normal.*

Proof. We prove for $f = f_i \| f_j \| \bar{f}_i \| \bar{f}_j$ since the proofs for other cases are similar. Since f_i is k-normal, f is at least k-normal. Also we observe that if $f_i = f_j$, then we have

$$f = f_i \| f_i \| \bar{f}_i \| \bar{f}_i.$$

Using the same technique in Theorem 5.4.6, we show the existence of a $(k+1)$-dimensional affine subspace where $f(\mathbf{x})$ is a constant. Let $\mathbf{z}_1, \ldots \mathbf{z}_k \in$ be k distinct, linearly independent vectors, $\mathbf{d} = (d_1, d_2, \ldots, d_{n-2})$ be a vector in $\mathbb{F}_2^{n-2}$, and $a_i \in \mathbb{F}_2$ be for $1 \leq i \leq k$. We define a k-dimensional flat $G = \{\mathbf{x} \in \mathbb{F}_2^{n-2} \mid \mathbf{x} = a_1\mathbf{z}_1 + a_2\mathbf{z}_2 + \cdots + a_k\mathbf{z}_k + \mathbf{d},\ a_i = \mathbb{F}_2,\ 1 \leq i \leq k\}$ such that $f_i|_G = 0$. In construction of f, we integrate two variables, x_{n-1} and x_n into the domain of f_i, and we can construct a $(k+1)$-dimensional flat in the following way. Let $\mathbf{z}_l = (z_{l1}, z_{l2}, \ldots, z_{l(n-2)})$ where $1 \leq l \leq k$. We set

$$\mathbf{z}'_l = (z_{l1}, z_{l2}, \ldots, z_{l(n-2)}, 0, 0),$$

$$\mathbf{z}'_{k+1} = (0, \ldots, 1, 0),$$

and

$$\mathbf{d}' = (d_1, d_2, \ldots, d_{n-2}, 0, 0)$$

where $\mathbf{z}'_{k+1},\ \mathbf{d}' \in \mathbb{F}_2^n$. Then

$$G' = \{\mathbf{x}' \in \mathbb{F}_2^n \mid \mathbf{x}' = a_1\mathbf{z}'_1 + a_2\mathbf{z}'_2 + \cdots + a_{k+1}\mathbf{z}'_{k+1} + \mathbf{d}',\ a_i = \mathbb{F}_2,\ 1 \leq i \leq k+1\}.$$

If a vector $\mathbf{x}' \in G'$ with $a_{k+1} = 0$, then f follows the first f_i in the construction. If a vector $\mathbf{x}' \in G'$ with $a_{k+1} = 1$, then f follows the second f_i in the construction. Therefore, G' is a $k+1$-dimensional flat such that $f\,|_{G'} = 0$. Therefore, the theorem holds. □

We also present a similar result on the normality of the functions of Construction 2. Let f_i in Construction 2 be k-normal but not $k+1$-normal, and we show that the function f of Construction 2 cannot have a constant function value on the $k+2$-dimensional flat $H = \{a_1\mathbf{e}_{i_1} \oplus \cdots \oplus a_{k+2}\mathbf{e}_{i_{k+2}} \oplus \mathbf{d}\}$, where $\mathbf{d} = (y_1, \ldots y_n)$ is a fixed vector in $\mathbb{F}_2^n$ and $\mathbf{e}_{i_m} = (x_1, \ldots, x_n)$ is an elementary vector such that $x_j = 1$ if and only if $j = i_m$ with $1 \leq i_m \leq n$. We assume $f = f_i \| f_j \| \bar{f}_i \| \bar{f}_j$, since the others can be shown by similar

arguments. Let us also assume that H exists. We observe that y_{i_m} is irrelevant (whether it is 0 or 1) due to $\mathbf{e}_{i_m}$, so we set $\mathbf{d}$ with $y_{i_1} = \ldots = y_{i_{k+2}} = 0$. To illustrate better, we rewrite the restriction of our function to H as follows:

$$\begin{aligned} f(\mathbf{x})\,|_H &= (\bar{x}_{n-1}f_i \oplus x_{n-1}f_j)\|(\bar{x}_{n-1}\bar{f}_i \oplus x_{n-1}\bar{f}_j)\,|_H \\ &= \bar{x}_n(\bar{x}_{n-1}f_i \oplus x_{n-1}f_j) \oplus x_n(\bar{x}_{n-1}\bar{f}_i \oplus x_{n-1}\bar{f}_j)\,|_H \\ &= \bar{x}_{n-1}(\bar{x}_n f_i \oplus x_n f_i) \oplus x_{n-1}(\bar{x}_n f_j \oplus x_n f_j) \oplus x_n\,|_H \\ &= f_i \oplus x_{n-1}f_i \oplus x_{n-1}f_j \oplus x_n\,|_H. \end{aligned}$$

Without loss of generality, we assume $f(\mathbf{x}) = 0$ for all $\mathbf{x} = (x_1, \ldots, x_n) \in H$, and we examine the following cases, depending upon the variables, x_{n-1} and x_n.

Case 1: $n-1,\ n \notin \{i_1, i_2, \ldots, i_{k+2}\}$. Then $x_{n-1} = d_{n-1}$, and $d_n = x_n$. We observe that, for all possible values for x_{n-1} and x_n, $f\,|_H$ follows one of the functions, f_i, $\bar{f}_i$, f_j, or $\bar{f}_j$. Since each function is only k-normal, there exists at least one $\mathbf{x} \in H$ with $f(\mathbf{x}) = 1$, a contradiction. We note that the other instances where $x_{n-1} = y_{i_m}$ or $x_n = y_{i_m}$ are covered by the other cases.

Case 2: $n-1 \notin \{i_1, i_2, \ldots, i_{k+2}\}$ and $x_n \in \{i_1, i_2, \ldots, i_{k+2}\}$. Then $x_{n-1} = d_{n-1}$. If $x_{n-1} = 0$, $f\,|_H$ follows the function, f_i or $\bar{f}_i$. We know each function is k-normal. Since f_i and $\bar{f}_i$ have opposite function values in H, there exists at least one $\mathbf{x} \in H$ with $f(\mathbf{x}) = 1$, a contradiction. If $x_{n-1} = 1$, $f\,|_H$ follows f_j, or $\bar{f}_j$, the same justification applies, and we have a contradiction.

Case 3: $n \notin \{i_1, i_2, \ldots, i_{k+2}\}$ and $x_{n-1} \in \{i_1, i_2, \ldots, i_{k+2}\}$. Then $d_n = x_n$. If $x_n = 0$, then $f\,|_H$ follows the function, $f_i \| f_j$. If $x_n = 1$, then $f\,|_H$ again follows the function, $\bar{f}_i \| \bar{f}_j$. In either case, we can only have a $k+1$-normal function, which is a contradiction.

Case 4: $x_{n-1},\, x_n \in \{i_1, i_2, \ldots, i_{k+2}\}$. In this case $f\,|_H$ follows $f = f_i \| f_i \| \bar{f}_i \| \bar{f}_i$. and any two vectors $\mathbf{x}',\, \mathbf{x}'' \in H$ in the forms of $\mathbf{x}' = (a_1, \ldots, a_{n-2}, 0, 0)$ and $\mathbf{x}'' = (b_1, \ldots, b_{n-2}, 0, 1)$ with $a_i,\, b_i \in \mathbb{F}_2$, $1 \leq i \leq n-2$ have opposite function values. Therefore, we have a contradiction.

Remark 5.4.8. References [73], and [74] contain the constructions of normal, or non-normal functions based upon some of the functions of Construction 1, namely $f_1||f_2||f_2||\bar{f}_1$, where f_i are bent or have some normality properties.

Finally, we investigate the propagation property of our construction.

Theorem 5.4.9. [27] *If the base functions f_1 and f_2 in Construction 1 satisfy the strict avalanche criterion, then f satisfies the strict avalanche criterion.*

Proof. We recall that we add two variables x_{n-1} and x_n when we concatenate the functions. For every vector $\mathbf{y} \in \mathbb{F}_2^n$, write $\mathbf{y} = (\mathbf{y}_{n-2}, y_{n-1}, y_n)$ with $\mathbf{y}_{n-2} \in \mathbb{F}_2^{n-2}$. We shall show the claim for $f = f_1||f_2||f_1||\bar{f}_2$, as all the other possibilities are similar. To apply Lemma 2.3.9, we check $f' = f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \mathbf{a})$ where $\mathbf{a} \in \mathbb{F}_2^n$ of weight $wt(\mathbf{a}) = 1$. We consider three possible cases.
Case 1. Let $\mathbf{a} = (0, \ldots, 0, 1)$. Then,

$$\begin{aligned} f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \mathbf{a}) &= (f_1||f_2)(\mathbf{x}_{n-2}, x_{n-1})\bar{x}_n \oplus (f_1||\bar{f}_2)(\mathbf{x}_{n-2}, x_{n-1})x_n \\ &\quad \oplus (f_1||f_2)(\mathbf{x}_{n-2}, x_{n-1})x_n \oplus (f_1||\bar{f}_2)(\mathbf{x}_{n-2}, x_{n-1})\bar{x}_n \\ &= (f_1||f_2)(\mathbf{x}_{n-2}, x_{n-1}) \oplus (f_1||\bar{f}_2)(\mathbf{x}_{n-2}, x_{n-1}) \\ &= 0_{2^{n-2}}||1_{2^{n-2}} \,\|\, 0_{2^{n-2}}||1_{2^{n-2}}, \end{aligned}$$

Clearly, it is a balanced function.

Case 2. Take $\mathbf{a} = (0, \ldots, 1, 0)$. Then

$$
\begin{aligned}
& f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \mathbf{a}) \\
= \; & (f_1||f_2)(\mathbf{x}_{n-2}, x_{n-1})\bar{x}_n \oplus (f_1||\bar{f}_2)(\mathbf{x}_{n-2}, x_{n-1})x_n \\
& \oplus (f_1||f_2)(\mathbf{x}_{n-2}, \bar{x}_{n-1})\bar{x}_n \oplus (f_1||\bar{f}_2)(\mathbf{x}_{n-2}, \bar{x}_{n-1})x_n \\
= \; & f_1(\mathbf{x}_{n-2})\bar{x}_{n-1}\bar{x}_n \oplus f_2(\mathbf{x}_{n-2})x_{n-1}\bar{x}_n \oplus f_1(\mathbf{x}_{n-2})\bar{x}_{n-1}x_n \oplus \bar{f}_2(\mathbf{x}_{n-2})x_{n-1}x_n \\
& \oplus f_1(\mathbf{x}_{n-2})x_{n-1}\bar{x}_n \oplus f_2(\mathbf{x}_{n-2})\bar{x}_{n-1}\bar{x}_n \oplus f_1(\mathbf{x}_{n-2})x_{n-1}x_n \oplus \bar{f}_2(\mathbf{x}_{n-2})\bar{x}_{n-1}x_n \\
= \; & f_1(\mathbf{x}_{n-2})\bar{x}_n \oplus f_2(\mathbf{x}_{n-2})\bar{x}_n \oplus f_1(\mathbf{x}_{n-2})x_n \oplus \bar{f}_2(\mathbf{x}_{n-2})x_n \\
= \; & f_1(\mathbf{x}_{n-2}) \oplus f_2(\mathbf{x}_{n-2}) \oplus x_n.
\end{aligned}
$$

which is balanced regardless of $f_1 \oplus f_2$ is balanced or not.

Case 3. Take $\mathbf{a} = (\mathbf{a}', 0, 0)$, with $wt(\mathbf{a}') = 1$. Write $\mathbf{x}_a = \mathbf{x}_{n-2} \oplus \mathbf{a}'$. Then,

$$
\begin{aligned}
& f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \mathbf{a}) \\
= \; & (f_1||f_2)(\mathbf{x}_{n-2}, x_{n-1})\bar{x}_n \oplus (f_1||\bar{f}_2)(\mathbf{x}_{n-2}, x_{n-1})x_n \\
& \oplus (f_1||f_2)(\mathbf{x}_a, x_{n-1})\bar{x}_n \oplus (f_1||\bar{f}_2)(\mathbf{x}_a, x_{n-1})x_n \\
= \; & f_1(\mathbf{x}_{n-2})\bar{x}_{n-1}\bar{x}_n \oplus f_2(\mathbf{x}_{n-2})x_{n-1}\bar{x}_n \oplus f_1(\mathbf{x}_{n-2})\bar{x}_{n-1}x_n \oplus \bar{f}_2(\mathbf{x}_{n-2})x_{n-1}x_n \\
& \oplus f_1(\mathbf{x}_a)\bar{x}_{n-1}\bar{x}_n \oplus f_2(\mathbf{x}_a)x_{n-1}\bar{x}_n \oplus f_1(\mathbf{x}_a)\bar{x}_{n-1}x_n \oplus \bar{f}_2(\mathbf{x}_a)x_{n-1}x_n \\
= \; & (f_1(\mathbf{x}_{n-2}) \oplus f_1(\mathbf{x}_a))\bar{x}_{n-1}\bar{x}_n \oplus (f_2(\mathbf{x}_{n-2}) \oplus f_2(\mathbf{x}_a))x_{n-1}\bar{x}_n \\
& \oplus (f_1(\mathbf{x}_{n-2}) \oplus f_1(\mathbf{x}_a))\bar{x}_{n-1}x_n \oplus (\bar{f}_2(\mathbf{x}_{n-2}) \oplus \bar{f}_2(\mathbf{x}_a))x_{n-1}x_n \\
= \; & (f_1(\mathbf{x}_{n-2}) \oplus f_1(\mathbf{x}_a))\bar{x}_{n-1} \oplus (f_2(\mathbf{x}_{n-2}) \oplus f_2(\mathbf{x}_a))x_{n-1},
\end{aligned}
$$

which is balanced. Since f_1 and f_2 satisfy the strict avalanche criterion, both $f_1(\mathbf{x}_{n-2}) \oplus f_1(\mathbf{x}_{n-2} \oplus \mathbf{a}')$ and $f_2(\mathbf{x}_{n-2}) \oplus f_2(\mathbf{x}_{n-2} \oplus \mathbf{a}')$ are balanced. We note that f' is balanced for all the cases. Then, we have

$$C_{\hat{f}}(\mathbf{u}) = 0,$$

for all $\mathbf{u} \in \mathbb{F}_2^n$ with $wt(\mathbf{u}) = 1$. By Lemma 2.3.9, we conclude that f satisfies the SAC. $\square$

Theorem 5.4.10. [27] *With* $\{i,\, j\} = \{1,\, 2\}$*, if* f_i, f_j *satisfy the strict avalanche criterion and* $f_i \oplus f_j$ *is balanced, then the functions of Construction* 2 *of the form* $f_i||f_j||\bar{f}_j||\bar{f}_i,\ \bar{f}_i||\bar{f}_j||f_j||f_i$ *satisfy the strict avalanche criterion.*

Proof. For every vector $\mathbf{y} \in \mathbb{F}_2^n$, we write $\mathbf{y} = (\mathbf{y}_{n-2}, y_{n-1}, y_n)$ with $\mathbf{y}_{n-2} \in \mathbb{F}_2^{n-2}$. We show the claim in the case $f = f_1||f_2||\bar{f}_2||\bar{f}_1$, as all the other possibilities are similar. Let $\mathbf{a} \in \mathbb{F}_2^n$ of weight $wt(\mathbf{a}) = 1$. We consider these three cases.

Case 1. Take $\mathbf{a} = (0, \ldots, 0, 1)$. Then

$$\begin{aligned} f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \mathbf{a}) &= (f_1||f_2)(\mathbf{x}_{n-2}, x_{n-1})\bar{x}_n \oplus (\bar{f}_2||\bar{f}_1)(\mathbf{x}_{n-2}, x_{n-1})x_n \\ &\quad \oplus (f_1||f_2)(\mathbf{x}_{n-2}, x_{n-1})x_n \oplus (\bar{f}_2||\bar{f}_1)(\mathbf{x}_{n-2}, x_{n-1})\bar{x}_n \\ &= (f_1||f_2)(\mathbf{x}_{n-2}, x_{n-1}) \oplus (\bar{f}_2||\bar{f}_1)(\mathbf{x}_{n-2}, x_{n-1}) \\ &= f_1(\mathbf{x}_{n-2}) \oplus f_2(\mathbf{x}_{n-2}) \oplus 1. \end{aligned}$$

Since $f_1(\mathbf{x}_{n-2}) \oplus f_2(\mathbf{x}_{n-2})$ is balanced, its complement is balanced.

Case 2. Take $\mathbf{a} = (0, \ldots, 1, 0)$. Then

$$
\begin{aligned}
& f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \mathbf{a}) \\
= \; & (f_1||f_2)(\mathbf{x}_{n-2}, x_{n-1})\bar{x}_n \oplus (\bar{f}_2||\bar{f}_1)(\mathbf{x}_{n-2}, x_{n-1})x_n \\
& \oplus (f_1||f_2)(\mathbf{x}_{n-2}, \bar{x}_{n-1})\bar{x}_n \oplus (\bar{f}_2||\bar{f}_1)(\mathbf{x}_{n-2}, \bar{x}_{n-1})x_n \\
= \; & f_1(\mathbf{x}_{n-2})\bar{x}_{n-1}\bar{x}_n \oplus f_2(\mathbf{x}_{n-2})x_{n-1}\bar{x}_n \oplus \bar{f}_2(\mathbf{x}_{n-2})\bar{x}_{n-1}x_n \oplus \bar{f}_1(\mathbf{x}_{n-2})x_{n-1}x_n \\
& \oplus f_1(\mathbf{x}_{n-2})x_{n-1}\bar{x}_n \oplus f_2(\mathbf{x}_{n-2})\bar{x}_{n-1}\bar{x}_n \oplus \bar{f}_2(\mathbf{x}_{n-2})x_{n-1}x_n \oplus \bar{f}_1(\mathbf{x}_{n-2})\bar{x}_{n-1}x_n \\
= \; & (f_1(\mathbf{x}_{n-2}) \oplus f_2(\mathbf{x}_{n-2}))\bar{x}_n \oplus (f_1(\mathbf{x}_{n-2}) \oplus f_2(\mathbf{x}_{n-2}))x_n, \\
= \; & f_1(\mathbf{x}_{n-2}) \oplus f_2(\mathbf{x}_{n-2}),
\end{aligned}
$$

which is balanced.

Case 3. Take $\mathbf{a} = (\mathbf{a}', 0, 0)$, with $wt(\mathbf{a}') = 1$. Write $\mathbf{x}_a = \mathbf{x}_{n-2} \oplus \mathbf{a}'$. Then,

$$f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \mathbf{a})$$

$$= (f_1||f_2)(\mathbf{x}_{n-2}, x_{n-1})\bar{x}_n \oplus (\bar{f}_2||\bar{f}_1)(\mathbf{x}_{n-2}, x_{n-1})x_n$$

$$\oplus(f_1||f_2)(\mathbf{x}_a, x_{n-1})\bar{x}_n \oplus (\bar{f}_2||\bar{f}_1)(\mathbf{x}_a, x_{n-1})x_n$$

$$= f_1(\mathbf{x}_{n-2})\bar{x}_{n-1}\bar{x}_n \oplus f_2(\mathbf{x}_{n-2})x_{n-1}\bar{x}_n \oplus \bar{f}_2(\mathbf{x}_{n-2})\bar{x}_{n-1}x_n \oplus \bar{f}_1(\mathbf{x}_{n-2})x_{n-1}x_n$$

$$\oplus f_1(\mathbf{x}_a)\bar{x}_{n-1}\bar{x}_n \oplus f_2(\mathbf{x}_a)x_{n-1}\bar{x}_n \oplus \bar{f}_2(\mathbf{x}_a)\bar{x}_{n-1}x_n \oplus \bar{f}_1(\mathbf{x}_a)x_{n-1}x_n$$

$$= (f_1(\mathbf{x}_{n-2}) \oplus f_1(\mathbf{x}_a))\bar{x}_{n-1}\bar{x}_n \oplus (f_2(\mathbf{x}_{n-2}) \oplus f_2(\mathbf{x}_a))x_{n-1}\bar{x}_n$$

$$\oplus(\bar{f}_2(\mathbf{x}_{n-2}) \oplus \bar{f}_2(\mathbf{x}_a))\bar{x}_{n-1}x_n \oplus (\bar{f}_1(\mathbf{x}_{n-2}) \oplus \bar{f}_1(\mathbf{x}_a))x_{n-1}x_n$$

$$= (f_1(\mathbf{x}_{n-2}) \oplus f_1(\mathbf{x}_a))(1 \oplus x_{n-1} \oplus x_n) \oplus (f_2(\mathbf{x}_{n-2}) \oplus f_2(\mathbf{x}_a))(x_{n-1} \oplus x_n)$$

$$= (f_1(\mathbf{x}_{n-2}) \oplus f_1(\mathbf{x}_a))\|(f_2(\mathbf{x}_{n-2}) \oplus f_2(\mathbf{x}_a))\|$$

$$(f_2(\mathbf{x}_{n-2}) \oplus f_2(\mathbf{x}_a))\|(f_1(\mathbf{x}_{n-2}) \oplus f_1(\mathbf{x}_a)).$$

Since f_1 and f_2 satisfy the strict avalanche criterion, both $f_1(\mathbf{x}_{n-2}) \oplus f_1(\mathbf{x}_a)$ and $f_2(\mathbf{x}_{n-2}) \oplus f_2(\mathbf{x}_a)$ are balanced. Therefore, f in Case 3 is balanced. Since f' is balanced for all the cases, we have

$$C_{\hat{f}}(\mathbf{u}) = 0,$$

for all $\mathbf{u} \in \mathbb{F}_2^n$ with $wt(\mathbf{u}) = 1$. By Lemma 2.3.9, we conclude that f satisfies the SAC. $\square$

6. AN APPLICATION OF THE TWO CONSTRUCTIONS

6.1. INTRODUCTION

In this chapter, we show an application of the construction methods presented in the previous chapter. In 2002, Krause [79] introduced an attack against stream ciphers based on the binary decision diagram (BDD). Several researchers have demonstrated the effectiveness of BDD-based attacks, and it has been difficult for functions with conventional cryptographic properties to counter BDD-based attacks. Various BDD-based attacks are found in [79], [80], [81], [82], and [83]. One way to counter BDD-based attacks is to integrate Boolean functions with robust BDDs [79]. There have been many constructions of Boolean functions with high algebraic immunity [77], [84], [85], [86], [87], [88], [89], [90], [91], [92], [93], [94], [95], [96], [97], [98], [99], but few took BDD-based attacks into consideration. In [100] and [101], Bryant showed that the hidden weighted-bit function (HWBF) has an exponential size of BDD regardless of variable order, and in [98], Wang et al. extensively investigated the cryptographic properties of HWBF. In this chapter, we briefly introduce the concept of the BDD and apply our construction methods from the previous chapter to HWBF. This chapter is based on Chung, Stanica, Tan, and Wang [27].

6.2. BINARY DECISION DIAGRAM (BDD)

We mention briefly relevant findings from [102, pp. 202–280], which covers BDDs extensively. Essentially, a BDD is a tree that represents a perspective on a Boolean function in which redundant nodes are removed. The BDD is an insightful way to represent a Boolean function, since it shows how the Boolean function data is stored and handled in a computer memory system [102, p. 202]. There are various BDD definitions in technical literature. Here, we assume the BDD has ordered vertices or nodes from the lowest at the top to the highest at the bottom, and is reduced as we apply the reduction steps explained below. We illustrate the BDD using an example from [102, pp. 202–205]. Let a Boolean

function, f, be described as in Table 6.1. A graphical way to represent the truth table f is using a tree structure shown Figure 6.1. We then apply a reduction algorithm on the tree, in which we remove nodes that represent a function also represented by another node in the BDD. Then we connect from the first x_2 to any 0 node and from the second x_2 to any 1 node. We note that two middle x_3 nodes have the same function values, so we combine them along with the edges from x_2 nodes, which results in a BDD representation of f in Table 6.2. A computer memory system can store f in four different memory blocks representing the nodes, and each block points to other nodes as indicated by the BDD [102, p. 203]. The size of the BDD, denoted by $BDD(f)$ is the number of vertices in a BDD.

$\mathbf{x} = x_1x_2x_3$	000	001	010	011	100	101	110	111
$f(\mathbf{x})$	0	0	0	1	0	1	1	1

Table 6.1: Truth Table of a Boolean Function f From [102, p. 205]

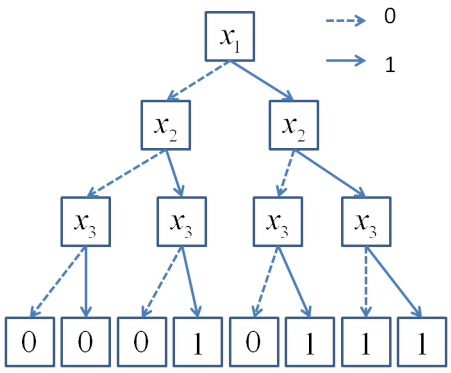


Figure 6.1: A Tree Representation of f

It is shown that every Boolean function has a unique BDD [102, p. 205]. The following are some benefits of considering BDD in Boolean function analysis [102, p. 206].

1. From the structural point of view, we can evaluate $f(\mathbf{x})$ in at most n steps by following the edges from the root vertex.

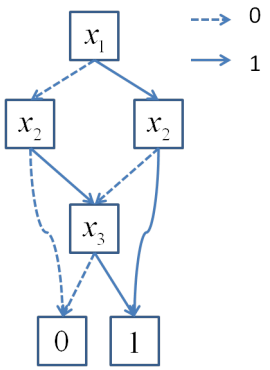


Figure 6.2: BDD Representation of f

2. We can effectively identify the lexicographically smallest $\mathbf{x}$ such that $f(\mathbf{x}) = 1$ or 0 in at most n steps.

3. We can find all $\mathbf{x} \in \mathbb{F}_2^n$ such that $f(\mathbf{x}) = 1$ or 0 in $O(BDD(f) \cdot n)$ steps.

4. We can efficiently generate random solutions to the equation $f(\mathbf{x}) = 1$ such that each solution gets generated in an equal probability.

5. We can solve the linear Boolean programming problem: Find $\mathbf{x} \in \mathbb{F}_2^n$ such that

$$u_1 x_1 \oplus u_2 x_2 \oplus \cdots \oplus u_n x_n = 1,$$

subject to

$$f(\mathbf{x}) = 1$$

with given constants $(u_1, u_2, \ldots, u_n)$ in $O(n + BDD(f))$ steps.

6.3. HIDDEN WEIGHTED-BIT FUNCTION (HWBF)

6.3.1. Definition of HWBF

In general, a HWBF h_n takes $\mathbf{x} = (x_n, x_{n-1}, \ldots, x_1)$ as input and outputs x_i , where $i = wt(\mathbf{x})$.

Definition 6.3.1. We define the HWBF of n variable, denoted by h_n as

$$h_n(\mathbf{x}) = \begin{cases} 0, & if\ wt(\mathbf{x}) = 0 \\ \\ x_{wt(\mathbf{x})}, & if\ wt(\mathbf{x}) > 0 \end{cases}$$

For example, we can evaluate $h_4(x_4, x_3, x_2, x_1)$ on $\mathbb{F}_2^4$ to obtain Table 6.2.

$x_4x_3x_2x_1$	$h_4(x_4, x_3, x_2, x_1)$	$x_4x_3x_2x_1$	$h_4(x_4, x_3, x_2, x_1)$
0000	0	1000	0
0001	1	1001	0
0010	0	1010	1
0011	1	1011	0
0100	0	1100	0
0101	0	1101	1
0110	1	1110	1
0111	1	1111	1

Table 6.2: A HWBF with $n = 4$

We observe that $h_4(0110) = 1$ since $wt(0110) = 2$ (so the second element of 0110 which is 1 is the function value). Table 6.3 has the list of HWBFs upto $n = 8$.

One of the interesting characteristics of HWBFs is that they have a very large number of nodes when represented by a BDD [79]. Specifically,

$$BDD(h_n) = c\chi^n + O(n^2),$$

where $\chi \approx 1.3247$ is the positive root of

$$\chi^3 = \chi + 1$$

and $c \approx 10.75115$ [102, p. 206].

6.3.2. Affine Structure within HWBF

In order to implement our construction methods with HWBFs, we need a class of functions affine equivalent to the HWBFs. It turned out that a HWBF h_n is, in fact, a concatenation of h_{n-1} and one of it affine-equivalent functions. Let ϕ be the left-rotation symmetric operation on vectors of arbitrary dimension, say $\phi(x_n, x_{n-1}, \ldots, x_1) = (x_1, \ldots, x_3, x_2)$.

n	HWBF of n Variable
1	01
2	0101
3	01010011
4	0101001100100111
5	01010011001001110010011000011111
6	01010011001001110010011000011111 00100110000111100000100101111111
7	01010011001001110010011000011111 00100110000111100000100101111111 00100110000111100000100101111110 00001000011010010001011111111111
8	01010011001001110010011000011111 00100110000111100000100101111111 00100110000111100000100101111110 00001000011010010001011111111111 00100110000111100000100101111110 00001000011010010001011111111110 00001000011010000001011011101001 00000001100101110111111111111111

Table 6.3: Hidden Weighted-Bit Functions

In [98], Wang et al. showed that the HWBF is a concatenation which can be iterated, as shown in the next formula,

$$h_n(x_1, \mathbf{x}, x_{n-1}, x_n) = h_{n-1}(x_1, \mathbf{x}, x_{n-1})||(h_{n-1} \circ \phi)(x_1, \mathbf{x}, x_{n-1})$$

$$= h_{n-2}(x_1, \mathbf{x})||(h_{n-2} \circ \phi)(x_1, \mathbf{x})||h_{n-2}(\mathbf{x}, x_{n-1})||(h_{n-2} \circ \phi)(\mathbf{x}, x_{n-1}) \tag{6.1}$$

$$= \cdots$$

where $\mathbf{x} = (x_2, \ldots, x_{n-2}) \in \mathbb{F}_2^{n-2}$. Noting this phenomenon, we define the function that describes the latter half of the HWBF.

Definition 6.3.2. Given the HWBF h_{n+1}, the *latter half function* of h_{n+1}, denoted by h'_n is

$$h'_n(\mathbf{x}) = \begin{cases} 1 \text{ if } wt(\mathbf{x}) = n \\ \\ x_{wt(\mathbf{x})+1} \text{ if } 0 \leq wt(\mathbf{x}) < n-1. \end{cases}$$

On the other hand, we call the other half, the *front half function*, which is h_{n-1}. So, we have

$$h_{n+1} = h_n \| h'_n.$$

6.3.3. Cryptographic Properties of HWBF

Wang et al. extensively investigated the cryptographic properties of HWBFs in [98]. We list their findings briefly. Given $h_n \in \mathcal{B}_n$ where h_n is an HWBF, the following statements are true:

- h_n is balanced.
- $\deg(h_n) = n-1$ for $n \geq 3$.
- h_n satisfies SAC.
- Let $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ and $wt(\mathbf{u}) = 1$. Then,

$$W_h(\mathbf{u}) \leq 4\binom{n-2}{\lceil \frac{n-2}{2} \rceil}. \tag{6.2}$$

- h_n has nonlinearity

$$nl(h_n) = 2^{n-1} - 2\binom{n-2}{\lceil \frac{n-2}{2} \rceil}.$$

- h_n has algebraic immunity

$$AI(h_n) \geq \left\lfloor \frac{n}{3} \right\rfloor + 1. \tag{6.3}$$

- h_n is a $\lfloor \frac{n}{2} \rfloor$-normal function, and h is not k-normal for any $k > \lfloor \frac{n}{2} \rfloor$.

Remark 6.3.3. We refer back to Table 6.3. We note the string of 1's at the end of the truth tables for each n. The pattern suggests that given $n \geq 5$, we may have at least last n bits to be 1. We ask if it is possible to exploit it. If an attack is possible, then what is the best way to mitigate the risk?

6.4. CONSTRUCTION BASED ON HWBF

For our constructions, we let $\{f_i, f_j\} = \{h_{n-2}, h'_{n-2}\}$. Then, we have,

Construction 1.

$$f_i \parallel f_j \parallel f_i \parallel \bar{f}_j;\ f_i \parallel f_j \parallel \bar{f}_i \parallel f_j;\ f_i \parallel \bar{f}_j \parallel f_i \parallel f_j;\ \bar{f}_i \parallel f_j \parallel f_i \parallel f_j;$$

$$f_i \parallel f_j \parallel f_j \parallel \bar{f}_i;\ f_i \parallel f_j \parallel \bar{f}_j \parallel f_i;\ f_i \parallel \bar{f}_j \parallel f_j \parallel f_i;\ \bar{f}_i \parallel f_j \parallel f_j \parallel f_i.$$

Construction 2.

$$f_i \parallel f_j \parallel \bar{f}_i \parallel \bar{f}_j;\ f_i \parallel f_j \parallel \bar{f}_j \parallel \bar{f}_i;\ \bar{f}_i \parallel \bar{f}_j \parallel f_i \parallel f_j;\ \bar{f}_i \parallel \bar{f}_j \parallel f_j \parallel f_i.$$

Theorem 6.4.1. [27] *Let $n \geq 4$ and $f_1 || f_2 = h_{n-2} \parallel h'_{n-2} = h_{n-1}$, the $(n-1)$- variables HWBF. Then, all of the functions f from Construction 1 are balanced of degree $\max\{n-2, 2\}$, have nonlinearity*

$$nl(f) = 2^{n-1} - 4\binom{n-4}{\lceil (n-4)/2 \rceil},$$

and have algebraic immunity

$$AI(f) \geq \left\lfloor \frac{n+2}{3} \right\rfloor.$$

Proof. Clearly, all functions in Construction 1 are balanced since h_{n-2} and h'_{n-2} are balanced. Furthermore, for any concatenation $g_1||g_2 \in \mathcal{B}_n$ where $g_1,\ g_2 \in \mathcal{B}_{n-1}$,

$$\deg(g_1||g_2) = \max\{\deg(g_1), \deg(g_1 \oplus g_2) + 1\}$$

since

$$\begin{aligned} g_1||g_2 &= (x_n \oplus 1)g_1 \oplus x_n g_2 \\ &= x_n(g_1 \oplus g_2) \oplus g_1. \end{aligned}$$

Thus,

$$\begin{aligned} \deg(f_1||f_2||f_1||\bar{f}_2) &= \max\{\deg(f_1||f_2), \deg((f_1||f_2) \oplus (f_1||\bar{f}_2)) + 1\} \\ &= \max\{n-2, \deg(0_{2^{n-2}}1_{2^{n-2}}) + 1\} \\ &= \max\{n-2, 2\}, \end{aligned}$$

where we write 0_s, or 1_s, for a truth table with the corresponding bit repeated s times.

Next, we do the computation for only one case. The others are similar. Let $f = f_1||f_2||f_1||\bar{f}_2$. We show that

$$\max|W_f(\mathbf{w})| = 8\binom{n-4}{\lceil \frac{n-4}{2} \rceil}.$$

We use Lemma 5.4.1, with $g_1 = h_{n-1} = f_1||f_2$, $g_2 = f_1||\bar{f}_2$, $f_1 = h_{n-2}$, and $f_2 = h'_{n-2}$. As in the proof of Theorem 5.4.4, we have

$$W_f(\mathbf{u}, u_{n-1}, u_n) \;=\; (1+(-1)^{u_n})\, W_{f_1}(\mathbf{u}) + (-1)^{u_{n-1}}\,(1-(-1)^{u_n})\, W_{f_2}(\mathbf{u})$$

where $\mathbf{u} \in \mathbb{F}_2^{n-2}$.

Thus,

$$W_f(\mathbf{u}, u_{n-1}, 0) = 2W_{f_1}(\mathbf{u})$$

and

$$W_f(\mathbf{u}, u_{n-1}, 1) = 2(-1)^{u_{n-1}} W_{f_2}(\mathbf{u}).$$

Since $f_1(\mathbf{u}) = h_{n-2}(\mathbf{u})$ and $f_2(\mathbf{u}) = h'_{n-2}(\mathbf{u})$ and $\max\limits_{\mathbf{u}\in\mathbb{F}_2^{n-2}} |W_{h_n}(\mathbf{u})| = 4\binom{n-4}{\lceil \frac{n-4}{2} \rceil}$ by Equation 6.2, it follows that

$$\begin{aligned} &\max_{(\mathbf{u},u_{n-1},u_n)\in\mathbb{F}_2^n} |W_f(\mathbf{u}, u_{n-1}, u_n)| \\ =\;& 2\max\left\{ \max_{\mathbf{u}\in\mathbb{F}_2^{n-2}} |W_{h_{n-2}}(\mathbf{u})|, \max_{\mathbf{u}\in\mathbb{F}_2^{n-2}} |W_{h_{n-2}}(\phi(\mathbf{u}))| \right\} = 8\binom{n-4}{\lceil \frac{n-4}{2} \rceil}. \end{aligned}$$

By Theorem 2.3.4, the nonlinearity of the functions in Construction 1 is

$$nl(f) = 2^{n-1} - 4\binom{n-4}{\lceil \frac{n-4}{2} \rceil}.$$

We now deal with the computation of the algebraic immunity for the considered functions. By Theorem 4 of [98], let

$$AI(h_n) = d_n \geq \left\lfloor \frac{n}{3} \right\rfloor + 1.$$

Since $h_n \sim h'_n$, we can construct an annihilator of h'_n by the same affine transformation between h_n and h'_n.

$$AI(h_n) = AI(h'_n).$$

By the definition of algebraic immunity,

$$AI(g) \quad = \quad AI(\bar{g})$$

for any Boolean function g, and also,

$$AI(f_i||f_j) \quad = \quad AI(f_j||f_i),$$

and by Lemma 5.4.3,

$$AI(f_i||\bar{f}_j) = AI(\bar{f}_i||f_j),$$

for $\{i,\, j\} = \{1,\, 2\}$.

So without loss of generality, we will only consider the case of $f = f_1||f_2||f_1||\bar{f}_2$. Let $g = g_1||g_2||k_1||k_2 \neq \mathbf{0}$ be a nonzero annihilator of f. Thus, g_1, k_1 are both annihilators of f_1; and, g_2, respectively, k_2 are annihilators of f_2, respectively, $\bar{f}_2$ such that each annihilator is a nonzero function.

First, since $g_1||g_2$ is an annihilator of $f_1||f_2 = h_{n-1}$, it follows that $\deg(g_1||g_2) = 0$, if both $g_1 = g_2 = 0$, or $\deg(g_1||g_2) \geq d_{n-1}$. Also, we observe that $\deg(g_1 \oplus k_1)$ is either 0, if $g_1 = k_1 = 0$ or $g_1 = k_1 \neq 0$. Otherwise, $\deg(g_1 \oplus k_1) \geq d_{n-1}$, since $g_1 \oplus k_1$ is an annihilator of f_1. Now, the degree of the concatenation $g = g_1||g_2||k_1||k_2$ is

$$\deg(g) = \max\{\deg(g_1||g_2), \deg((g_1 \oplus k_1)||(g_2 \oplus k_2)) + 1\}.$$

Next, we analyze the components of the set above. We see that

$$\deg(g_1||g_2) = \max\{\deg(g_1), \deg(g_1 \oplus g_2) + 1\},$$

and

$$\deg((g_1 \oplus k_1)||(g_2 \oplus k_2)) = \max\{\deg(g_1 \oplus k_1), \deg(g_1 \oplus g_2 \oplus k_1 \oplus k_2) + 1\}.$$

If we minimize $\max\{\deg(g_1 \oplus k_1), \deg(g_1 \oplus g_2 \oplus k_1 \oplus k_2) \oplus 1\}$, we have the worst case when $g_1 = k_1$ and $g_2 = k_2$. Then,

$$\deg(g) = \max\{\deg(g_1||g_2), 1\} \geq \left\lfloor \frac{n-1}{3} \right\rfloor + 1 = \left\lfloor \frac{n+2}{3} \right\rfloor$$

by Equation 6.3. □

Theorem 6.4.2. [27] *Let $n \geq 3$ and $f_1||f_2 = h_{n-1}$, the $(n-1)$-variables HWBF. All of the functions f from Construction 2 are balanced, have degree $n-2$, have nonlinearity*

$$nl(f) = 2^{n-1} - 4\binom{n-3}{\lceil (n-3)/2 \rceil},$$

have algebraic immunity

$$AI(f) \geq \left\lfloor \frac{n+2}{3} \right\rfloor,$$

and have the resiliency of order 1.

Proof. The functions in Construction 2 are balanced regardless of the balancedness of f_1 and f_2 and their complements. We will consider only some cases, since the others follow similarly. If there is a noteworthy difference, we will point it out as necessary. Let $f = f_1||f_2||\bar{f}_1||\bar{f}_2$. Clearly,

$$\begin{aligned}
\deg(f_1||f_2||\bar{f}_1||\bar{f}_2) &= \max\{\deg(f_1||f_2), \deg((f_1||f_2) \oplus (\bar{f}_1||\bar{f}_2)) + 1\} \\
&= \max\{n-2, \deg(0_{2^{n-1}}) + 1\} \\
&= \max\{n-2, 1\}. \\
&= n-2
\end{aligned}$$

for $n \geq 3$. For the other possibilities, if $f = f_1||f_2||\bar{f}_2||\bar{f}_1$,

$$\begin{aligned}
\deg(f_1||f_2||\bar{f}_2||\bar{f}_1) &= \max\{\deg(f_1||f_2), \deg((f_1||f_2) \oplus (\bar{f}_2||\bar{f}_1)) + 1\} \\
&= \max\{n-2, \deg((f_1 \oplus \bar{f}_2)||(f_2 \oplus \bar{f}_1)) + 1\} \\
&= \max\{n-2, \deg(f_1 \oplus \bar{f}_2) + 1\} \\
&= \max\{n-2, n-2\} \\
&= n-2.
\end{aligned}$$

Next, by Lemma 5.4.1 with $g_1 = h_{n-1} = f_1||f_2$, $g_2 = \bar{f}_1||\bar{f}_2$, $f_1 = h_{n-2}$, and $f_2 = h'_{n-2}$, as in Theorem 5.4.4 we have

$$\begin{aligned} W_f(\mathbf{u}, u_{n-1}, u_n) &= (1-(-1)^{u_n})(W_{f_1}(\mathbf{u}) + (-1)^{u_{n-1}} W_{f_2}(\mathbf{u})) \\ &= (1-(-1)^{u_n})\, W_{h_{n-1}}(\mathbf{u}, u_{n-1}), \end{aligned}$$

where $\mathbf{u} \in \mathbb{F}_2^{n-2}$. We now get

$$\max_{(\mathbf{u},u_{n-1},u_n)\in\mathbb{F}_2^n} |W_f(\mathbf{u}, u_{n-1}, u_n)| = 8\binom{n-3}{\lceil \frac{n-3}{2} \rceil}$$

by Equation 6.2. Therefore, we have

$$nl(f) = 2^{n-1} - 4\binom{n-3}{\lceil \frac{n-3}{2} \rceil}$$

by Theorem 2.3.4.

To show resilience of order 1, we will prove that the functions in Construction 2 are correlation immune of order 1 since the function is already balanced. The case of $f_1||f_2||\bar{f}_1||\bar{f}_2$, or $\bar{f}_1||\bar{f}_2||f_1||f_2$, is straightforward. Let $f = f_1||f_2||\bar{f}_2||\bar{f}_1$. To show correlation immunity of order 1, we need to show that $W_f(\mathbf{w}) = 0$ for any vector $\mathbf{w}$ with $wt(\mathbf{w}) = 1$ by Lemma 2.3.15. It turns out that this will follow simply by using the balancedness of f_1 and f_2 and not the HWBF property. By Lemma 5.4.1, if $wt(\mathbf{u}, u_{n-1}, u_n) = 1$, we have

$$W_f(\mathbf{u}, u_{n-1}, u_n) = \left(1-(-1)^{u_{n-1}+u_n}\right)(W_{f_1}(\mathbf{u}) + (-1)^{u_{n-1}} W_{f_2}(\mathbf{u})).$$

Now, if $wt(u_{n-1}, u_n) = 1$, then $\mathbf{u} = 0$. Since f_1 and f_2 are balanced,

$$W_{f_1}(\mathbf{u}) = W_{f_2}(\mathbf{u}) = 0.$$

If $wt(u_{n-1}, u_n) = 0$, we have

$$1 - (-1)^{u_{n-1}+u_n} = 0.$$

Therefore,

$$W_f(\mathbf{u}, u_{n-1}, u_n) = 0,$$

where $wt(wt(\mathbf{u}, u_{n-1}, u_n) = 1$, and the functions have the resiliency of order 1.

The computation of the algebraic immunity is similar to the one in the proof of Theorem 6.4.1. Let $f = f_1||f_2||\bar{f}_1||\bar{f}_2$. We see that

$$AI(f_1||f_2) = AI(\bar{f}_1||\bar{f}_2).$$

Additionally, by the definition of algebraic immunity, the annihilator used to justify the AI of $f_1||f_2$ or $\bar{f}_1||\bar{f}_2$ can be the same function. Let $g = g_1||g_2 \neq \mathbf{0}$ be a nonzero annihilator of f where $g_1,\ g_2 \in \mathcal{B}_{n-1}$. The degree of the concatenation $g = g_1||g_2$ is

$$\deg(g) = \max\{\deg(g_1), \deg(g_1 \oplus g_2) \oplus 1\}.$$

We observe that this value takes a minimum when $g_1 = g_2$. So we have

$$\begin{aligned}
\min\{\deg(g)\} &= \min\{\max\{\deg(g_1), \deg(g_1 \oplus g_2) \oplus 1\}\} \\
&= \deg(g_1) \\
&= \left\lfloor \frac{n-1}{3} \right\rfloor + 1 \\
&= \left\lfloor \frac{n+2}{3} \right\rfloor
\end{aligned}$$

by Equation 6.3, which gives us $AI(f) \geq \left\lfloor \frac{n+2}{3} \right\rfloor$. □

We see that Theorems 5.4.6 and 5.4.7 apply to the normality of the Construction 1 and 2 functions, respectively.

Example 6.4.3. We present a snapshot of a performance comparison between the base function HWBF and a function of Construction 1. Let $f = f_1 \parallel f_2 \parallel f_1 \parallel \bar{f}_2$. In Table 6.4, one can find the algebraic immunity and nonlinearity of f, compared to the HWBF h_n.

n	$\mathcal{AI}(f)$	$\mathcal{AI}(h)$	$nl(f)$	$nl(h_n)$
7	3	3	52	44
8	4	4	104	88
9	4	4	216	186
10	5	4	432	372
11	5	5	884	772
12	5	5	1768	1544
13	6	5	3592	3172
14	6	5	7184	6344
15	6	6	14536	12952

Table 6.4: Algebraic immunity and nonlinearity of the HWBF-based f and the HWBF h From [27]

As for the algebraic immunity, let $fg = h_n$, $\deg(g) = d$ and $\deg(h_n) = e$. In Table 6.5, we present the lowest possible values of (d, e) needed for the fast algebraic attack.

n	7	8	9	10	11	12	13
(d, e)	(1,3)	(1,5)	(1,5)	(1,7)	(1,7)	(1,9)	(1,9)
	(2,4)	(2,4)	(2,4)	(2,5)	(2,6)	(2,8)	(2,8)
	(3,3)	(3,4)	(3,4)	(3,5)	(3,5)	(3,6)	(3,6)
				(4,5)	(4,5)	(4,6)	(4,6)
							(5,6)

Table 6.5: Behavior of the HWBF-based function f against Fast Algebraic Attacks From [27]

Remark 6.4.4. We briefly mention some tentative results on our constructions with the Carlet–Feng function. Let $f_1 \in \mathcal{B}_{10}$ be the Carlet–Feng function with the primitive poly-

nomial

$$x^{10} + x^6 + x^5 + x^3 + x^2 + x + 1$$

and $f_2(\mathbf{x}) = f_1(A\mathbf{x})$, where

$$A = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3, \mathbf{e}_4, \mathbf{e}_5, \mathbf{e}_{10}, \mathbf{e}_6, \mathbf{e}_7, \mathbf{e}_8, \mathbf{e}_9)$$

and $\mathbf{e}_i \in \mathbb{F}_2^{10}$ is the unit column vector with 1 on the i-th position and 0's elsewhere. Let $f = f_1||f_2||f_1||\bar{f}_2 \in \mathcal{B}_{12}$. Then, we computed $AI(f) = 6$ and $nl(f) = 1992$. In comparison, the nonlinearity of the 12-variable Carlet-Feng function discussed in [96] and [97] is only 1970. Also, the recent 12-variable functions constructed by Construction 1 and 2 of [96] have the nonlinearity at most 1988 and 1982, respectively. Our constructions compare well to competitive constructions with good cryptographic properties.

7. CONCLUSION AND FUTURE RESEARCH

7.1. CONCLUSION

In this dissertation, we studied the affine equivalence of Boolean functions, the relationship between Boolean functions and graphs, and the construction techniques of Boolean functions and their applications. Affine equivalence of Boolean functions still remains a tough challenge for researchers. We defined S-equivalence, a special type of affine equivalence based on permutation of variables, and our research focused on S-equivalence of MRS functions and circulant matrices of $\mathbb{F}_2$. We established a relationship between MRS functions and the circulant matrices of $\mathbb{F}_2$. We explored the group structure of the circulant matrices of $\mathbb{F}_2$ and found a pattern of the square of a circulant matrix of $\mathbb{F}_2$. This pattern ultimately helped us to a series of properties of MRS functions of which circulant matrices are singular, but have pseudo inverses. We showed a condition in terms of generating polynomials for a singular circulant matrix in $\mathbb{F}_2$ to have a general or reflexive inverse. We defined a dual function for an MRS function with respect to the inverse of the circulant matrix of the function. We then showed that two S-equivalent functions have the same degree in ANF, and their dual functions have the same degree. We also showed that if two MRS functions of which circulant matrices are P-Q equivalent, they have the same degree. Moreover, if the matrices are invertible, their dual functions have the same degree, and a circulant matrix of one of the original functions is a permutation of the other.

We developed an idea to represent an MRS function in a graph using the cycles generated by the ordered short algebraic normal form (OSANF) of the function. We illustrated that this graph is regular. We showed that the graph is ultimately determined by the sequential differences of the indices of variables in OSANF. We described the relationship between this property and the construction of MRS functions.

We considered two effective constructions of cryptographic Boolean functions, which use a base function with strong cryptographic properties, one of its affine equivalent func-

tions, and simple construction techniques, namely complementation and concatenation. This strategy reinforces the two important requirements for cryptographic functions, namely security and speed. Security is clearly a must requirement. However, if a cryptographic function requires an unreasonable amount of computing power or hard-to-implement hardware or software, it cannot be utilized effectively. We presented an application of the constructions, using hidden weighted-bit functions.

In summary, we cleared some trenches on the way to a complete understanding of the affine-equivalence problem of Boolean functions. We further presented two effective constructions for cryptographic Boolean functions.

7.2. FUTURE WORK

In this dissertation, we explored various areas of Boolean functions. We solved some related problems in the process, but we could not solve all the problems. We present a partial list of problems worth considering.

1. Prove or disprove "*If* $f \overset{s}{\sim} g$ *with singular matrices* A_f *and* A_g*, and* $wt(\Delta(f)) = wt(\Delta(g))$*, then* $wt(\Delta(f^\dagger)) = wt(\Delta(g^\dagger))$*, where* $f^\dagger$ *and* $g^\dagger$ *are pseudoinverses of* f *and* g*, respectively*".

2. We propose a thorough analysis of the CCGs. More graph-theoretic, number-theoretic, and combinatorial analyses can be done. One can also study the relationship between the CCG and cryptographic properties. One can expand the concept of CCG and develop a CCG-like structure for all RSBFs.

3. Extend the cryptographic analysis of Constructions 1 and 2 to GAC,..., etc. Study more applications of the constructions using other functions.

4. The BDD of Boolean functions has an interesting set of operations. Their effects on various cryptographic properties of Boolean functions would be a worthwhile project.

5. HWBFs seem to display predictable patterns in the second half of a truth table. An interesting project will be to engineer another class of cryptographic Boolean functions with high BBD size, but without the predictability.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] K. Kim, "Construction of DES-like S-boxes based on Boolean functions satisfying the SAC," in *ASIACRYPT '91*, ser. Lecture Notes in Computer Science, vol. 739. Springer Verlag, 1993, pp. 59–72.

[2] NIST, "DATA ENCRYPTION STANDARD (DES)," http://www.itl.nist.gov/fipspubs/fip46-2.htm, Mar. 2013.

[3] J. Pieprzyk and C. X. Qu, "Rotation-symmetric functions and fast hashing," *Journal of Universal Computer Science*, vol. 5, no. 1, pp. 20–31, 1999.

[4] T. Cusick and P. Stanica, *Cryptographic Boolean Functions and Applications*. Academic Press, Elsevier, 2009.

[5] J. L. Shafer, S. W. Schneider, J. T. Butler, and P. Stanica, "Enumeration of bent Boolean functions by reconfigurable computer," in *FCCM '10*. IEEE Computer Society, 2010, pp. 265–272.

[6] J.-H. Evertse, "Linear structures in blockciphers," in *Advances in Cryptology, EUROCRYPT'87*, ser. Lecture Notes in Computer Science, D. Chaum and W. L. Price, Eds. Springer Berlin Heidelberg, 1988, vol. 304, pp. 249–266.

[7] X. Lai, "Additive and linear structures of cryptographic functions," in *Fast Software Encryption*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1995, vol. 1008, pp. 75–85.

[8] J. D. Golic and G. Morgari, "On the resynchronization attack," in *Fast Software Encryption*, ser. Lecture Notes in Computer Science, T. Johansson, Ed. Springer Berlin Heidelberg, 2003, vol. 2887, pp. 100–110.

[9] X. Zhang and Y. Zheng, "GAC - the criterion for global avalanche characteristics of cryptographic functions," *Journal of Universal Computer Science*, vol. 1, pp. 316–333, 1995.

[10] B. Schneier, "Description of a new variable-length key, 64-bit block cipher (Blowfish)," in *Fast Software Encryption '93, Cambridge Security Workshop*, ser. Lecture Notes in Computer Science, vol. 809. Springer-Verlag, 1994, pp. 191–204.

[11] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "On the twofish key schedule," in *Selected Areas in Cryptography*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1999, vol. 1556, pp. 27–42.

[12] Wikipedia, "Turing (cipher)," http://http://en.wikipedia.org/wiki/Turing_(cipher), Mar. 2013.

[13] G. G. Rose and P. Hawkes, "Turing: A fast stream cipher," in *Fast Software Encryption '03*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, vol. 2887, pp. 290–306.

[14] C. E. Shannon, "Communication theory of secrecy systems," *Bell Systems Technical Journal*, vol. 28, pp. 656–715, 1949.

[15] W. Millan, A. Clark, and E. Dawson, "Smart hill climbing finds better Boolean functions," in *Workshop on Selected Areas in Cryptology 1997*, 1997, pp. 50–63.

[16] T. Jakobsen and L. R. Knudsen, "The interpolation attack on block ciphers," in *Proceedings of the 4th International Workshop on Fast Software Encryption*, ser. FSE '97. Springer-Verlag, 1997, pp. 28–40.

[17] S. Moriai, T. Shimoyama, and T. Kaneko, "Higher order differential attack using chosen higher order differences," in *Proceedings of the Selected Areas in Cryptography*, ser. SAC '98. Springer-Verlag, 1999, pp. 106–117.

[18] J. Seberry and X.-M. Zhang, "Highly nonlinear 0-1 balanced Boolean functions satisfying strict avalanche criterion (extended abstract)," in *Advances in Cryptology - AUSCRYPT '92*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1993, vol. 718, pp. 143–155.

[19] Q. Wang and T. Johansson, "A note on fast algebraic attacks and higher order nonlinearities," in *Information Security and Cryptology '11*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 6584, pp. 404–414.

[20] W. Meier and O. Staffelbach, "Fast correlation attacks on stream ciphers," in *Advances in Cryptology-EUROCRYPT'88*, ser. Lecture Notes in Computer Science, vol. 330. New York, NY, USA: Springer-Verlag New York, Inc., 1988, pp. 301–314.

[21] A. F. Webster and S. E. Tavares, "On the design of S-boxes," in *Advances in cryptology–CRYPTO 85*, ser. Lecture Notes in Computer Science, vol. 218. New York, NY, USA: Springer-Verlag New York, Inc., 1986, pp. 523–534.

[22] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, and J. Vandewalle, "Propagation characteristics of Boolean functions," in *EUROCRYPT '90*, ser. Lecture Notes in Computer Science, vol. 473. New York, NY, USA: Springer-Verlag, 1991, pp. 161–173.

[23] T. Siegenthaler, "Correlation-immunity of nonlinear combining functions for cryptographic applications (corresp.)," *Information Theory, IEEE Transactions on*, vol. 30, no. 5, pp. 776 – 780, Sep. 1984.

[24] N. Courtois, "Fast algebraic attacks on stream ciphers with linear feedback," in *Advances in Cryptology - CRYPTO'03*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, vol. 2729, pp. 176–194.

[25] N. T. Courtois and W. Meier, "Algebraic attacks on stream ciphers with linear feedback," in *EUROCRYPT'03*, ser. Lecture Notes in Computer Science, vol. 2656. Springer, 2003.

[26] W. Meier, E. Pasalic, and C. Carlet, "Algebraic attacks and decomposition of Boolean functions," in *In Advances in Cryptology - EUROCRYPT 2004*, ser. Lecture Notes in Computer Science, vol. 3027. Springer-Verlag, 2004, pp. 474–491.

[27] J. Chung, P. Stanica, C. H. Tan, and Q. Wang, "Construction of Boolean functions with good cryptographic properties," 2012, submitted Manuscript.

[28] P. Hawkes and G. Rose, "Rewriting variables: The complexity of fast algebraic attacks on stream ciphers," in *Advances in Cryptology - CRYPTO 2004*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2004, vol. 3152, pp. 390–406.

[29] H. Dobbertin, "Construction of bent functions and balanced Boolean functions with high nonlinearity," in *Fast Software Encryption*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1995, vol. 1008, pp. 61–74.

[30] A. Canteaut, M. Daum, H. Dobbertin, and G. Leander, "Finding nonnormal bent functions," *Discrete Appl. Math.*, vol. 154, no. 2, pp. 202–218, Feb. 2006.

[31] C. Carlet, "On cryptographic complexity of Boolean functions," in *Finite Fields with Applications to Coding Theory, Cryptography and Related Areas*, G. L. Mullen, H. Stichtenoth, and H. Tapia-Recillas, Eds. Springer Berlin Heidelberg, 2002, pp. 53–69.

[32] P. Charpin, "Normal Boolean functions," *J. Complex.*, vol. 20, no. 2-3, pp. 245–265, Apr. 2004.

[33] Y. V. Tarannikov, "On resilient Boolean functions with maximal possible nonlinearity," in *Progress in Cryptology, INDOCRYPT 2000*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2000, vol. 1977, pp. 19–30.

[34] C. Carlet, "On the coset weight divisibility and nonlinearity of resilient and correlation-immune functions," in *Sequences and their Applications*, ser. Discrete Mathematics and Theoretical Computer Science. Springer London, 2002, pp. 131–144.

[35] M. Lobanov, "Tight bound between nonlinearity and algebraic immunity," *IACR Cryptology ePrint Archive*, vol. 2005, p. 441, 2005. [Online]. Available: http://eprint.iacr.org/2005/441

[36] M. A. Harrison, "On the classification of Boolean functions by the general linear and affine groups," *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, no. 2, pp. 285–299, 1964.

[37] E. Berlekamp and L. Welch, "Weight distributions of the cosets of the (32,6) Reed-Muller code," *Information Theory, IEEE Transactions on*, vol. 18, no. 1, pp. 203–207, 1972.

[38] J. A. Maiorana, "A classification of the cosets of the Reed-Muller code r(1,6)," *Math. Comp. 57 (1991)*, pp. 403–414, 1991.

[39] N. Patterson and D. Wiedemann, "The covering radius of the Reed-Muller code is at least 16276," *IEEE Trans. Inf. Theor.*, vol. 29, no. 3, pp. 354–356, Sep. 2006.

[40] S. Kavut, S. Maitra, S. Sarkar, and M. D. Yücel, "Enumeration of 9-variable rotation symmetric Boolean functions having nonlinearity > 240," in *INDOCRYPT'06*, ser. Lecture Notes in Computer Science, vol. 4329. Springer, 2006, pp. 266–279.

[41] S. Kavut, S. Maitra, and M. D. Yücel, "Search for Boolean functions with excellent profiles in the rotation symmetric class," *IEEE Transactions on Information Theory*, vol. 53, no. 5, pp. 1743–1751, 2007.

[42] S. Kavut and M. D. Yücel, "Generalized rotation symmetric and dihedral symmetric Boolean functions - 9 variable Boolean functions with nonlinearity 242," in *AAECC '07*, ser. Lecture Notes in Computer Science, vol. 4851. Springer, 2007, pp. 321–329.

[43] E. Filiol and C. Fontaine, "Highly nonlinear balanced Boolean functions with a good correlation-immunity," in *EUROCRYPT'98*, ser. Lecture Notes in Computer Science, vol. 1403. Springer, 1998, pp. 475–488.

[44] J. A. Clark, J. L. Jacob, S. Maitra, and P. Stanica, "Almost Boolean functions: The design of Boolean functions by spectral inversion," *Computational Intelligence*, vol. 20, no. 3, pp. 450–462, 2004.

[45] M. Hell, A. Maximov, and S. Maitra, "On efficient implementation of search strategy for rotation symmetric Boolean functions," in *In Ninth International Workshop on Algebraic and Combinatoral Coding Theory 2004*, 2004.

[46] P. Stanica, S. Maitra, and J. A. Clark, "Results on rotation symmetric bent and correlation immune Boolean functions," in *FSE '04*, ser. Lecture Notes in Computer Science, vol. 3017. Springer, 2004, pp. 161–177.

[47] A. Maximov, "Classes of plateaued rotation symmetric Boolean functions under transformation of Walsh spectra," *IACR Cryptology ePrint Archive*, vol. 2004, p. 354, 2004. [Online]. Available: http://eprint.iacr.org/2004/354

[48] P. Stanica and S. Maitra, "Rotation symmetric Boolean functions - count and cryptographic properties," *Discrete Applied Mathematics*, vol. 156, no. 10, pp. 1567–1580, 2008.

[49] S. Fu, C. Li, K. Matsuura, and L. Qu, "Construction of rotation symmetric Boolean functions with maximum algebraic immunity," in *Cryptology and Network Security*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5888, pp. 402–412.

[50] S. Su and X. Tang, "Construction of rotation symmetric Boolean functions with optimal algebraic immunity and high nonlinearity," *Designs, Codes and Cryptography*, pp. 1–17, 2012.

[51] A. Brown and T. W. Cusick, "Equivalence classes for cubic rotation symmetric functions," *Cryptography and Communications*, vol. 5, no. 2, pp. 85–118, 2013.

[52] M. L. Bileschi, T. W. Cusick, and D. Padgett, "Weights of Boolean cubic monomial rotation symmetric functions," *Cryptography Commun.*, vol. 4, no. 2, pp. 105–130, Jun. 2012.

[53] T. Cusick and Y. Cheon, "Affine equivalence for rotation symmetric Boolean functions with 2^k variables," *Designs, Codes and Cryptography*, vol. 63, pp. 273–294, 2012.

[54] T. W. Cusick and A. Brown, "Affine equivalence for rotation symmetric Boolean functions with $\mathrm{p^k}$ variables," *Finite Fields and Their Applications*, vol. 18, no. 3, pp. 547–562, 2012.

[55] T. W. Cusick, "Affine equivalence of cubic homogeneous rotation symmetric functions," *Inf. Sci.*, vol. 181, no. 22, pp. 5067–5083, Nov. 2011.

[56] J. Chung and P. Stanica, "On the affine equivalence of monomial rotation symmetric Boolean functions," 2013, submitted Manuscript.

[57] Y. Zhang and Y. Deng, "Results on permutation symmetric Boolean functions," *Journal of Systems Science and Complexity*, vol. 26, no. 2, pp. 302–312, 2013.

[58] P. J. Davis, *Circulant Matrices*. John Wiley and Sons, New York, 1979.

[59] A. Ben-Israel and T. N. Greveille, *Generalized Inverses*. New York, NY, USA: Springer-Verlag, 2003.

[60] M. Pearl, "generalized inverses of matrices with entries taken from an arbitrary field," *Linear Algebra and its Applications*, vol. 1, 1968.

[61] D. Bini, G. M. D. Corso, G. Manzini, and L. Margara, "Inversion of circulant matrices over $\mathbb{Z}_m$," *Math. Comput.*, vol. 70, no. 235, pp. 1169–1182, 2001.

[62] R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications*. New York, NY, USA: Cambridge University Press, 1986.

[63] D. Wiedemann, Michael, and E. Zieve, "Equivalence of sparse circulants: The bipartite ADAM problem," http://dept.math.lsa.umich.edu/~zieve/papers/circulants.pdf, 2007.

[64] A. Bernasconi and B. Codenotti, "Spectral analysis of Boolean functions as a graph eigenvalue problem," *IEEE Trans. Comput.*, vol. 48, no. 3, pp. 345–351, Mar. 1999.

[65] A. Bernasconi, B. Codenottl, and J. Vanderkam, "A characterization of bent functions in terms of strongly regular graphs," *Computers, IEEE Transactions on*, vol. 50, no. 9, pp. 984–985, Sep. 2001.

[66] P. Stanica, "Graph eigenvalues and Walsh spectrum of Boolean functions," *Integers*, vol. 7, 2007.

[67] R. Yarlagadda and J. E. Hershey, "Analysis and synthesis of bent sequences," *IEEE Proceedings (Computers and Digital Techniques)*, vol. 136, no. 2, pp. 112–123, 1989.

[68] J. Dillon, "Elementary Hadamard difference sets," Ph.D. dissertation, University of Maryland, 1974.

[69] R. L. McFarland, "A family of difference sets in non-cyclic groups," *J. Comb. Theory, Ser. A*, vol. 15, no. 1, pp. 1–10, 1973.

[70] C. Carlet, "Two new classes of bent functions," in *Advances in Cryptology EUROCRYPT 93*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1994, vol. 765, pp. 77–101.

[71] ——, "A construction of bent functions," *Finite Fields and Applications, London Mathematical Society, Lecture Series 233*, pp. 47–58, 1996.

[72] ——, "On the secondary constructions of resilient and bent functions," in *Coding, Cryptography and Combinatorics*, ser. Progress in Computer Science and Applied Logic. Birkhauser Basel, 2004, vol. 23, pp. 3–28.

[73] C. Carlet, H. Dobbertin, and G. Leander, "Normal extensions of bent functions," *Information Theory, IEEE Transactions on*, vol. 50, no. 11, pp. 2880–2885, 2004.

[74] S. Gangopadhyay and D. Sharma, "On construction of non-normal Boolean functions," *The Australasian Journal of Combinatorics*, vol. 38, pp. 267–272, 2007.

[75] E. Pasalic, "A design of Boolean functions resistant to (fast) algebraic cryptanalysis with efficient implementation," *Cryptography and Communications*, vol. 4, no. 1, pp. 25–45, 2012.

[76] W. Wang, M. Liu, and Y. Zhang, "Comments on design of Boolean functions resistant to (fast) algebraic cryptanalysis with efficient implementation," *Cryptography and Communications*, vol. 5, pp. 1–6, 2013.

[77] C. Carlet, D. Dalai, K. Gupta, and S. Maitra, "Algebraic immunity for cryptographically significant Boolean functions: analysis and construction," *Information Theory, IEEE Transactions on*, vol. 52, no. 7, pp. 3105 – 3121, Jul. 2006.

[78] M. Liu, Y. Zhang, and D. Lin, "Perfect algebraic immune functions," in *Proceedings of the 18th international conference on The Theory and Application of Cryptology and Information Security*, ser. ASIACRYPT'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 172–189.

[79] M. Krause, "BDD-based cryptanalysis of keystream generators," in *Advances in Cryptology EUROCRYPT 2002*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2002, vol. 2332, pp. 222–237.

[80] M. Krause and D. Stegemann, "Reducing the space complexity of BDD-based attacks on keystream generators," in *Fast Software Encryption*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, vol. 4047, pp. 163–178.

[81] Y. Shaked and A. Wool, "Cryptanalysis of the bluetooth e_0 cipher using OBDDs," in *ISC '06*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, vol. 4176, pp. 187–202.

[82] M. Krause, "OBDD-based cryptanalysis of oblivious keystream generators," *Theory of Computing Systems*, vol. 40, pp. 101–121, 2007.

[83] D. Stegemann, "Extended BDD-based cryptanalysis of keystream generators," in *Selected Areas in Cryptography 2007*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, vol. 4876, pp. 17–35.

[84] A. Braeken and B. Preneel, "On the algebraic immunity of symmetric Boolean functions," in *INDOCRYPT'05*, ser. Lecture Notes in Computer Science, vol. 3797. Springer Verlag, 2005, pp. 35–48.

[85] C. Carlet and K. Feng, "An infinite class of balanced functions with optimal algebraic immunity, good immunity to fast algebraic attacks and good nonlinearity," in *Advances in Cryptology - ASIACRYPT 2008*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, vol. 5350, pp. 425–440.

[86] ——, "An infinite class of balanced vectorial Boolean functions with optimum algebraic immunity and good nonlinearity," in *Coding and Cryptology*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5557, pp. 1–11.

[87] D. K. Dalai, K. C. Gupta, and S. Maitra, "Cryptographically significant Boolean functions: Construction and analysis in terms of algebraic immunity," in *Fast Software Encryption*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, vol. 3557, pp. 98–111.

[88] D. K. Dalai, S. Maitra, and S. Sarkar, "Basic theory in construction of Boolean functions with maximum possible annihilator immunity," *Des. Codes and Cryptography*, vol. 40, no. 1, pp. 41–58, Jul. 2006.

[89] N. Li and W.-F. Qi, "Construction and analysis of Boolean functions of 2t+1 variables with maximum algebraic immunity," in *Advances in Cryptology, ASIACRYPT 2006*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, vol. 4284, pp. 84–98.

[90] N. Li, L. Qu, W.-F. Qi, G. Feng, C. Li, and D. Xie, "On the construction of Boolean functions with optimal algebraic immunity," *IEEE Trans. Inf. Theor.*, vol. 54, no. 3, pp. 1330–1334, Mar. 2008.

[91] E. Pasalic, "Almost fully optimized infinite classes of Boolean functions resistant to (fast) algebraic cryptanalysis," in *Information Security and Cryptology, ICISC 2008*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5461, pp. 399–414.

[92] E. Pasalic and Y. Wei, "On the construction of cryptographically significant Boolean functions using objects in projective geometry spaces," *Information Theory, IEEE Transactions on*, vol. 58, no. 10, pp. 6681–6693, Oct. 2012.

[93] L. Qu, K. Feng, F. Liu, and L. Wang, "Constructing symmetric Boolean functions with maximum algebraic immunity," *Information Theory, IEEE Transactions on*, vol. 55, no. 5, pp. 2406–2412, May 2009.

[94] P. Rizomiliotis, "On the resistance of Boolean functions against algebraic attacks using univariate polynomial representation," *Information Theory, IEEE Transactions on*, vol. 56, no. 8, pp. 4014–4024, Aug 2010.

[95] C. H. Tan and S.-T. Goh, "Several classes of even-variable balanced Boolean functions with optimal algebraic immunity," *IEICE Transactions*, vol. 94-A, no. 1, pp. 165–171, 2011.

[96] D. Tang, C. Carlet, and X. Tang, "Highly nonlinear Boolean functions with optimal algebraic immunity and good behavior against fast algebraic attacks," *Information Theory, IEEE Transactions on*, vol. 59, no. 1, pp. 653–664, Jan 2013.

[97] Z. Tu and Y. Deng, "A conjecture about binary strings and its applications on constructing Boolean functions with optimal algebraic immunity," *Designs, Codes and Cryptography*, vol. 60, pp. 1–14, 2011.

[98] Q. Wang, C. Carlet, P. Stanica, and C. Tan, "Cryptographic properties of the hidden weighted bit function," 2012, submitted manuscript.

[99] X. Zeng, C. Carlet, J. Shan, and L. Hu, "More balanced Boolean functions with optimal algebraic immunity and good nonlinearity and resistance to fast algebraic attacks," *Information Theory, IEEE Transactions on*, vol. 57, no. 9, pp. 6310–6320, Sep. 2011.

[100] R. E. Bryant, "On the complexity of VLSI implementations and graph representations of Boolean functions with application to integer multiplication," *Computers, IEEE Transactions on*, vol. 40, no. 2, pp. 205–213, Feb. 1991.

[101] ——, "Symbolic Boolean manipulation with ordered binary-decision diagrams," *ACM Computing Surveys*, vol. 24, pp. 293–318, 1992.

[102] D. E. Knuth, *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms Part 1*, 1st ed. Addison-Wesley Professional, 2008.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California